Spring 5-16-2020

Application of Optimal Switching Using Adaptive Dynamic Programming in Power Electronics

Ataollah Gogani Khiabani
Southern Methodist University, atagktc@gmail.com

APPLICATION OF OPTIMAL SWITCHING USING ADAPTIVE DYNAMIC PROGRAMMING IN POWER ELECTRONICS

Approved by:

Dr. Ali Heydari
Assistant Professor of Mechanical Engineering

Dr. Yildirim Hurmuzlu
Professor of Mechanical Engineering

Dr. Edmond Richer
Associate Professor of Mechanical Engineering

Dr. Mohammad Khodayar
Associate Professor of Electrical Engineering

Dr. Dario Villarreal Suarez
Adjunct Professor of Electrical Engineering

APPLICATION OF OPTIMAL SWITCHING USING ADAPTIVE DYNAMIC PROGRAMMING IN POWER ELECTRONICS

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Mechanical Engineering

by

Ataollah Gogani Khiabani

B.Sc., Electrical Engineering, University of Tabriz
M.Sc., Electrical Engineering, K. N. Toosi University of Technology

May 16, 2020

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my supervisor Dr. Ali Heydari for his support and guidance. His invaluable help and encouragement inspired me in my work. I also appreciate the support of my PhD advisory committee.

Gogani Khiabani, Ataollah B.Sc., Electrical Engineering, University of Tabriz
M.Sc., Electrical Engineering, K. N. Toosi University of Technology

Application of Optimal Switching Using Adaptive Dynamic Programming in Power Electronics

Advisor: Dr. Ali Heydari
Doctor of Philosophy degree conferred May 16, 2020
Dissertation completed May 16, 2020

In this dissertation, optimal switching in switched systems using adaptive dynamic programming (ADP) is presented. Two applications in power electronics, namely single-phase inverter control and permanent magnet synchronous motor (PMSM) control are studied using ADP. In both applications, the objective of the control problem is to design an optimal switching controller, which is also relatively robust to parameter uncertainties and disturbances in the system.

An inverter is used to convert the direct current (DC) voltage to an alternating current (AC) voltage. The control scheme of the single-phase inverter uses a single function approximator, called critic, to evaluate the optimal cost and determine the optimal switching. After offline training of the critic, which is a function of system states and elapsed time, the resulting optimal weights are used in online control, to get a smooth output AC voltage in a feedback form. Simulations show the desirable performance of this controller with linear and nonlinear load and its relative robustness to parameter uncertainty and disturbances. Furthermore, the proposed controller is upgraded so that the inverter is suitable for single-phase variable frequency drives. Finally, as one of the few studies in the field of adaptive dynamic programming (ADP), the proposed controllers are implemented on a physical prototype to show the performance in practice.

The torque control of PMSMs has become an interesting topic recently. A new approach based on ADP is proposed to control the torque, and consequently the speed of a PMSM when an unknown load torque is applied on it. The proposed controller achieves a fast transient response, low ripples and small steady-state error. The control algorithm uses two neural networks, called critic and

actor. The former is utilized to evaluate the cost and the latter is used to generate control signals. The training is done once offline and the calculated optimal weights of actor network are used in online control to achieve fast and accurate torque control of PMSMs. This algorithm is compared with field-oriented control (FOC) and direct torque control based on space vector modulation (DTC-SVM). Simulations and experimental results show that the proposed algorithm provides desirable results under both accurate and uncertain modeled dynamics.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF TABLES xi

CHAPTER

1. INTRODUCTION 1
   - 1.1. Adaptive Dynamic Programming 1
   - 1.2. Switched Systems 2
   - 1.3. Single-Phase Inverter 2
     - 1.3.1. Inverter Switching Problem Formulation 4
   - 1.4. Permanent Magnet Synchronous Motor Control 5
     - 1.4.1. PMSM Control Problem Formulation 7
   - 1.5. Structure of the Thesis 9
2. Design and implementation of an optimal switching controller for uninterruptible power supply inverters using adaptive dynamic programming 10
   - 2.1. Introduction 10
   - 2.2. Single Phase Voltage Source Inverter 13
   - 2.3. Optimal Switching Using ADP 14
   - 2.4. Implementation on Voltage Source Inverter 17
     - 2.4.1. Neural Network Training 17
     - 2.4.2. Performance With Different Loads 20
     - 2.4.3. Sensitivity to System Parameters 22
   - 2.5. Single Phase Variable Frequency Drive 25
   - 2.6. Experimental Results 30
   - 2.7. Conclusion 33
3. Preliminaries on Permanent Magnet Synchronous Motor Control 38

3.1. Clark and Park Transform 38
3.2. Permanent Magnet Synchronous Motor 40
3.3. Space Vector Pulse Width Modulation 43
3.4. Field-Oriented Control 45
3.5. Direct Torque Control 46
3.6. Direct Torque Control with Space Vector Modulation 51
4. Optimal Torque Control of Permanent Magnet Synchronous Motors Using Adaptive Dynamic Programming 54
4.1. Introduction 54
4.2. Dynamics of PMSM 57
4.3. Optimal Control Using ADP 58
4.4. Implementation on PMSM 62
4.4.1. Neural Network Training 62
4.4.2. Comparative simulation 66
4.5. Experimental results 69
4.5.1. Comparative experiment under nominal condition 70
4.5.2. Comparative experiment with parameter variations 73
4.6. Conclusion 75
5. Conclusions and Future Work 78
5.1. Conclusions 78
5.2. Future Work 78
APPENDIX
A. Motor Control Algorithms Using Adaptive Dynamic Programming 80
B. Torque Sensor Datasheet 82
BIBLIOGRAPHY 85

LIST OF FIGURES

Figure		Page
1.1	Single phase voltage source inverter circuit (picture adapted from [7]).	3
1.2	Three phase voltage source inverter connected to motor.	6
2.1	Single phase voltage source inverter circuit (picture adapted from [7]).	13
2.2	History of weights during learning iterations for fixed amplitude and frequency	19
2.3	Output voltage and output current for a linear resistive load	20
2.4	Active modes for a linear resistive load	21
2.5	Output voltage of linear resistive load with different initial conditions for ADP (Fig. 5.a) and FCS-MPC (Fig. 5.b)	22
2.6	Output voltage, current and voltage error of nonlinear load	23
2.7	Output voltage and current for linear resistive load with a step change from no load to $R_L = 30\Omega$	24
2.8	Output voltage of linear resistive load when there is a mismatch in system parameters	25
2.9	A. Output voltage of the feedback controller under changing input DC voltage B. Output voltage of the open-Loop controller under changing input DC voltage	26
2.10	Output voltage for variable frequency and amplitude when only states and time are fed to the critic network	27
2.11	History of weights during learning iterations for variable amplitude, frequency, and phase angle	29
2.12	Output voltage for variable frequency and amplitude when states, time, amplitude, frequency and phase angle are fed to the critic network	30
2.13	Implemented circuit of the single phase inverter	31
2.14	Experimental result of the output voltage for a linear resistive load with THD=0.3%	34
2.15	Experimental result of the output voltage for a nonlinear load with THD=0.35%	34
2.16	Experimental result of the output current for a nonlinear load	35

2.17 Experimental result of the output voltage for no-load to full-load 35
2.18 Experimental result of the output voltage of ADP controller with parameter mismatch 36
2.19 Experimental result of the output voltage of PWM controller with parameter mismatch 36
2.20 Experimental result of the output voltage of ADP controller with variable frequency and amplitude 37
3.1 Current space vector and its projections 39
3.2 transformation of three-phase currents to stationary two-phase currents. (Picture adapted from [73]) 40
3.3 dq rotating frame 41
3.4 transformation of stationary two-phase currents to rotating dq current. (Picture adapted from [73]) 42
3.5 surface-mount PMSM and Interior PMSM, (Picture adapted from [29]) 42
3.6 Basic voltage vectors and a reference voltage vector in a complex plane, (Picture adapted from [15]) 44
3.7 Pulse pattern of voltage vector located in sector 1 45
3.8 Block Diagram of the FOC with SVM, (Picture adapted from [15]) 47
3.9 Load angle and torque adjustment by voltage vector. (Picture adapted from [75]) 48
3.10 Voltage vector options to cause fast dynamics for either increasing or decreasing the developed machine torque. (Picture adapted from [75]) 49
3.11 Flux linkage band and consecutive voltage vectors in DTC (Picture adapted from [75]) 50
3.12 DTC block diagram (Picture adapted from [75]) 51
3.13 Hysteresis controllers: (a) torque controller and (b) flux linkage controller. (Picture adapted from [75]) 52
3.14 DTC-SVM block diagram (Picture adapted from [75]) 53
4.1 PMSM control block diagram using actor neural network 63
4.2 History of critic weights during learning iterations 66
4.3 Speed response simulation of three controllers when a load torque is applied on the motor at rated speed 67

4.4 Torque response simulation of three controllers when a load torque is applied on the motor at the rated speed of 3000 rpm 68

4.5 Speed response simulation of three controllers when a load torque is applied on the motor at rated speed with parameter uncertainties 70

4.6 PMSM control experimental setup 71

4.7 Speed response experiment of three controllers when a load of 0.7 $N.m$ is applied at 2000 rpm 72

4.8 Torque response experiment of three controllers when a load of 0.7 $N.m$ is applied at 2000 rpm 73

4.9 Speed response experiment of three controllers under parameter uncertainties when a step load of 0.7 $N.m$ is applied at 2000 rpm 74

4.10 torque response experiment of three controllers under parameter uncertainties when a step load of 0.7 $N.m$ is applied at 2000 rpm 75

4.11 Speed response experiment of three controllers when motor is under the load of 0.7 $N.m$ 76

4.12 Speed response experiment of three controllers under parameter uncertainties when motor is under the load of 0.7 $N.m$ 77

LIST OF TABLES

Table		Page
2.1	Switch states of the unipolar single phase inverter	15
2.3	Parameters of the single phase inverter shown in Fig. 2.1 (adapted from [43].)	18
2.5	Parameters of the implemented single phase VSI shown in Fig. 2.1	32
3.1	Voltage vectors generated by a two-level VSI	43
3.3	SVM sequence and timing	45
3.5	DTC Switching Table	52
4.1	Motor and Control System Parameters	64
4.3	ITAE of motor torque when a load torque of 0.6 $N.m$ is applied after one second on the motor at the rated speed of 3000 rpm	69
4.5	Motor Parameters with uncertainties	69
4.7	ITAE of motor torque and speed when a load torque of 0.7 $N.m$ is applied at 2000 rpm	72
4.9	ITAE of motor torque and speed under parameter uncertainties when a load torque of 0.7 $N.m$ is applied at 2000 rpm	76

To Yasaman for her endless love and support.

Chapter 1

INTRODUCTION

Optimal control is an approach to determine the control decisions that will cause a process to satisfy the physical constraints and at the same time minimize (or maximize) some performance criterion [38]. One systematic approach to solve the optimal control problem is dynamic programming. Dynamic programming is based on Bellman's principle of optimality, which states that [47]:

An optimal policy has the property that no matter what the previous decision (controls) have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions.

Dynamic programming approach incorporates two components. The first one, is the discrete-time dynamic model of the system and the second component is a cost function, which depends on system states and, possibly, decisions at each state. Dynamic programming solves the problem offline and backward in time. This solution is usually saved in look-up tables and is used in online control. The major disadvantage of dynamic programming is the so-called curse of dimensionality [38]. This issue refers to the fact that as the problem size increases, the size of system states and decisions increase, which leads to exponential growth in computational load. This exponential growth quickly overwhelms the computational resources and makes it difficult or impossible to solve the problem in reasonable time.

1.1. Adaptive Dynamic Programming

In order to deal with the challenges of dynamic programming, another approach called adaptive (approximate) dynamic programming (ADP) or reinforcement learning (RL) is proposed in literature [4, 22, 27, 31, 39, 46, 48, 52, 71, 88]. Reinforcement learning is the modification of actions based on interactions with the environment [58]. In ADP, the idea behind RL is utilized to solve the optimal control problem. ADP typically uses two neural networks called critic and actor, to solve

the optimal control problem. The critic network approximates the optimal cost-to-go and the actor network approximates the optimal policy. ADP is inherently an iterative approach; therefore, the optimal cost-to-go and optimal policy are obtained iteratively. There are two well-known iterative approaches in the literature, which are used in ADP, called value iteration and policy iteration. Value iteration (VI) starts with a guess on initial value function, updates the policy with the value function, uses a recursion to find the next value function and so on. On the other hand, policy iteration (PI), starts with a stabilizing policy, solves a Lyapunov equation to find the corresponding value function, updates the optimal policy and so on. Both VI and PI have advantages and disadvantages. For instance, in order to find the value function at each step, PI solves a Lyapunov equation, which is referred to as full backup, however VI uses a recursion, which is referred to as partial backup. Therefore, the computational load of PI is higher than VI. On the other hand, since PI starts the algorithm with a stabilizing policy, all the evolving policies under PI are stabilizing. However, this is not the case for VI generally. The research in [26], proposes some conditions that give VI the property of having evolving stabilizing policies like PI.

1.2. Switched Systems

A switched system is comprised of several modes (subsystems) with different dynamics such that at each time instant, only one mode can be active. In order to control such systems, one should determine not only the time of the switching, but also the mode to switch to [27]. ADP has shown great promises in solving optimal control problem in switched systems. Both VI and PI algorithms are used in the literature to control switched systems [25, 66, 78]. As mentioned before, since VI leads to lower computational load compared to PI, in this study, we will use VI to solve optimal switching problems.

1.3. Single-Phase Inverter

In recent years, fossil fuel limitations, climate changes, and economic restrictions have increased the need for other sources of energy. Distributed generation (DG) systems based on renewable energy sources such as wind turbines, photovoltaics, and micro-turbines have proved to be a decent

substitution for other sources of energy. The output voltage of DG sources are generally direct current (DC) and this voltage needs to be inverted to alternating current (AC) voltage since electrical transmission and loads typically need AC power, [63]. Another application of DC to AC conversion is in uninterruptible power supply (UPS) units. These units are generally used to supply high quality output with desired amplitude and frequency to critical loads such as life support systems, data storage units and telecommunication systems. Control of UPS must be capable to ensure that the output voltage has a low total harmonic distortion (THD) with negligible steady state error. Also it is crucial that the inverter be stable under both linear and nonlinear loads. Finally, since the system parameters usually change as the system ages, relative robustness to parameter uncertainties is a key factor in designing the inverter controller, [43]. One of the most common inverter topologies is voltage source inverter (VSI). VSI is a switched circuit and a common method of controlling it is the Pulse Width Modulation (PWM) technique. There has been different PWM-based control methods in literature [1, 21, 32, 34, 49, 59, 85]. However, our method is not PWM-based and has variable switching frequency.

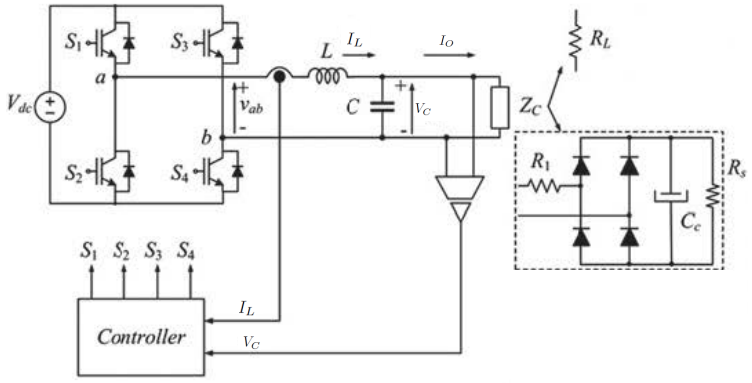


Figure 1.1: Single phase voltage source inverter circuit (picture adapted from [7]).

There are some challenges with single-phase inverter control, which we aim at solving them. The first challenge is the switching loss in the system. In PWM-based approaches, switching frequency is usually fixed, therefore in order to get a desirable output with low THD, high switching frequency is chosen which leads to high losses. We want to use ADP to design an inverter with

variable switching frequency and reach a low THD output with lower number of switching and hence, lower losses.

Another challenge with inverters is that, system parameters change after some time. Therefore, the controller should be relatively robust to parameter uncertainty. This issue is considered in this work and ADP is used to solve this problem.

A single phase UPS inverter is shown in Fig. 1.1. It consists of a power stage, an output LC filter and the load, which is denoted by Z_C. By suitable switching of four switches $\{S_1,S_2,S_3,S_4\}$, we aim at inverting the input DC voltage to a desired output AC voltage. The desired output AC voltage is defined as $V_{out}(t) = V_m \sin(wt)$, where w is the angular frequency and t is time. The aim of the optimal controller is to determine the appropriate switching such that, the capacitor voltage (V_C) tracks the desired output voltage.

Two cases are considered for the inverter output. First, a controller is designed to convert a DC voltage to an AC voltage with a specified amplitude and frequency. Secondly, the controller is modified such that the user can choose the amplitude and frequency of the output on the fly.

1.3.1. Inverter Switching Problem Formulation

Let us define the switched system with M modes with the following nonlinear equation

$$x_{k+1} = f_i(x_k), k \in \mathbb{Z}_+, i \in \mathscr{I}, \tag{1.1}$$

where $x_k \in \mathbb{R}^n$ is the state vector at discrete time instances shown by non-negative integer k. Subscript i represents one of the M modes of the system and the system dynamics are given by $f_i : \mathbb{R}^n \to \mathbb{R}^n, \forall i \in \mathscr{I} := \{1,2,...,M\}$. The discrete cost function is defined as:

$$J = \sum_{k=0}^{\infty} \gamma^k Q(x_k, x_{des_k}) \tag{1.2}$$

where $\gamma \in (0,1]$ is the discount factor which is used to ensure the boundedness of the cost. $Q : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ is a continuous, positive definite function, which penalizes the tracking error between actual system states and desired system states. The optimal value function is defined as

the cost-to-go from current state at current time instant, if optimal decisions are made from current instant onwards. Using the definition of optimal value function we can write (1.2) as follows

$$V^*(x_0, x_{des_0}) := \sum_{k=0}^{\infty} \gamma^k Q(x_k^*, x_{des_k}), \tag{1.3}$$

where $x_0^* := x_0$ and x_k^*, $\forall k = 0, 1, 2, \ldots$ is the optimal trajectory under optimal switching. If the optimal value function is obtained, then at each time instant, the optimal mode can be selected. In Chapter 2, value iteration is utilized to find the optimal value function and hence, the optimal mode. The contents of Chapter 2 are published in [36].

1.4. Permanent Magnet Synchronous Motor Control

Permanent Magnet Synchronous Motors (PMSMs) are increasingly used in variable speed industrial drives. In contrast to induction motors, the PMSMs do not have the copper loss associated with rotor in induction motors. This copper loss reduction significantly improves the efficiency of PMSMs compared to induction motors. The high efficiency and high torque-inertia ratio of the variable speed PMSMs, along with high power density and low rotor losses have made these motors the preferred solution in industry [64]. In order to control the speed, torque or position of the PMSMs, a power electronic converter interfaces the power supply and motor [37, 90, 93]. As seen in Fig. 1.2, the three-phase two-level inverter has 6 switched. There are 8 combinations for these switches in which six of them are active, which means they produce a non-zero voltage on motor terminals, and two combinations lead to zero voltages.

Field oriented control (FOC) and direct torque control (DTC) are the two main control approaches of alternate current (AC) servo drives. A typical FOC scheme consists of two inner current loops and one outer speed loop. Proportional-integral (PI) controllers are commonly used to regulate the motor currents. In order to avoid large overshoots, the bandwidth of these current controllers is limited, which leads to the slow dynamic response of the motor, [54]. Furthermore, the PI gains play a crucial role in the steady state behavior of the motor under FOC, therefore fine tuning of the gains is necessary [5, 51]. In order to control a PMSM through power inverters, typically space

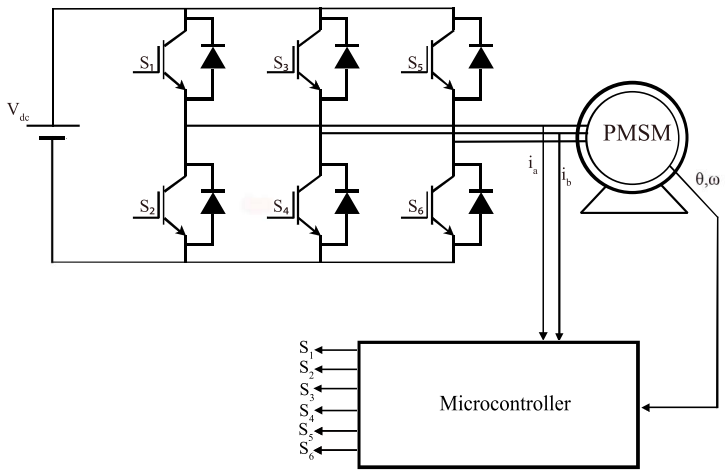


Figure 1.2: Three phase voltage source inverter connected to motor.

vector modulation (SVM) is used to realize the appropriate voltage, which is applied to the motor. This method can provide voltage vectors with adjustable amplitude and phase, [75].

DTC, on the other hand, utilizes another approach for the control. In classical DTC, based on two hysteresis comparators and a switching table, a suitable voltage vector is applied on the motor for the whole sampling time [91]. Although DTC provides a faster dynamic response compared to FOC, it has major disadvantages such as increased torque and stator flux ripples, [54]. Furthermore, DTC requires high sampling frequency for digital implementation of the controller, which in turn, demands more powerful digital signal processors (DSPs) and increases the cost of implementation [62]. In recent years, there has been many studies to address the disadvantages of the classical DTC, [77]. One of these approaches utilizes DTC with SVM. Unlike DTC, which uses one of the available fixed voltage vectors, with a fixed amplitude and phase, for the duration of the control cycle, any arbitrary voltage vector can be generated and applied to the motor, when DTC is augmented with SVM (i.e., DTC-SVM). This voltage vector, which has an adjustable amplitude and phase, is generated using multiple vectors during the sampling time. This approach leads to a reduction in torque and flux ripples [33, 44, 45].

FOC has a desirable steady state behavior, on the other hand DTC has a better transient response compared to FOC. The main issue with DTC is the torque ripples during steady state condition. In this thesis, ADP framework is used to design a controller which leads to desirable performance for both transient response and steady-state response. Also an important objective of this work is achieving good torque control under parameter uncertainty.

1.4.1. PMSM Control Problem Formulation

Compared to the inverter control, which only needs the critic network, the PMSM control utilizes both critic and actor networks. Therefore, the general optimal control formulation is presented in this subsection. Interested readers are referred to [22] for a more complete derivation of this approach.

Let the infinite-horizon cost function, subject to minimization, be given by

$$J = \sum_{k=0}^{\infty} \gamma^k (Q(x_k) + u_k^T R u_k), \tag{1.4}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the system states with dimension n, and the control vector with dimension m, respectively. Moreover, $Q : \mathbb{R}^n \to \mathbb{R}_+$ penalizes the states and $R \in \mathbb{R}^{m \times m}$ penalizes the control. Furthermore, $\gamma \in (0,1]$ is the discount factor, which is used to ensure the boundedness of the cost. The discrete-time nonlinear dynamics is defined as

$$x_{k+1} = f(x_k) + g(x_k)u_k, k \in \{0,1,2,...\}, x(0) = x_0, \tag{1.5}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function which represents the internal dynamics of the system and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$, is the input gain function. The objective is to find the sequence of 'optimal' control, denoted with $u_k^*, k \in \{0,1,2,...\}$ such that the cost function in (1.4) is minimized subject to system dynamics in (1.5).

One can write the cost-to-go from current time to infinity, as a function of current state and future decisions, denoted with $V(.,.)$, as

$$V(x_k, \{u_h\}_{h=k}^{\infty}) = \sum_{h=k}^{\infty} \gamma^k (Q(x_h) + u_h^T R u_h). \tag{1.6}$$

It is possible to write the above equation in a recursive form as

$$\begin{aligned} V(x_k, \{u_h\}_{h=k}^{\infty}) &= Q(x_k) + u_k^T R u_k + \gamma V(x_{k+1}, \{u_h\}_{h=k+1}^{\infty}) \\ &= Q(x_k) + u_k^T R u_k + \gamma V(f(x_k) + g(x_k)u_k, \{u_h\}_{h=k+1}^{\infty}), \end{aligned} \tag{1.7}$$

Function $V^*(x_k)$ is called the *value function* which is the optimal cost-to-go from current state at current time to infinity, if the optimal control sequence is applied on the system. Considering the relation given by (1.7), one can find the value function and optimal control sequence based on Bellman principle of optimality [38] as follows

$$\begin{aligned} V^*(x_k) &= inf_{\{u_h\}_{h=k}^{\infty}} (V(x_k, \{u_h\}_{h=k}^{\infty})) \\ &= inf_{u_k \in \mathbb{R}^m} \Big(Q(x_k) + u_k^T R u_k + \gamma V^*(f(x_k) + g(x_k)u_k) \Big), \forall x_k \in \mathbb{R}^n, \end{aligned} \tag{1.8}$$

$$u^*(x_k) = arginf_{u_k \in \mathbb{R}^m} \Big(Q(x_k) + u_k^T R u_k + \gamma V^*(f(x_k) + g(x_k)u_k) \Big), \forall x_k \in \mathbb{R}^n, \tag{1.9}$$

which leads to

$$u^*(x) = -\frac{1}{2} \gamma R^{-1} g(x)^T \nabla V^*(f(x) + g(x)u^*(x)), \forall x \in \mathbb{R}^n, \tag{1.10}$$

$$V^*(x) = Q(x) + u^*(x)^T R u^*(x) + \gamma V^*(f(x) + g(x)u^*(x)), \forall x \in \mathbb{R}^n, \tag{1.11}$$

Therefore, the optimal control and optimal value function can be obtained using (1.10) and (1.11). In Chapter 4, value iteration is used to solve this control problem.

1.5. Structure of the Thesis

The contents of this thesis are organized in five chapters. Chapter 1 corresponds to a brief introduction of this research and the applications that are studied in this work. The problem formulations are also presented in this chapter. Chapter 2 delves into the problem of optimal control of an inverter. Simulations and experimental results are presented in this chapter. Some preliminaries on PMSM control such as space vector pulse width modulation (SVPWM), field-oriented control (FOC) and direct torque control (DTC) are presented in Chapter 3. Chapter 4 presents the optimal control of a PMSM and compares this approach to FOC and DTC methods both in simulations and experiments. Finally, in Chapter 5 the main conclusions of this thesis, as well as some potential future research are pesented.

Chapter 2

Design and implementation of an optimal switching controller for uninterruptible power supply inverters using adaptive dynamic programming

A new approach based on adaptive dynamic programming is proposed to control single phase uninterruptible power supply inverters. The control scheme uses a single function approximator, called critic, to evaluate the optimal cost and determine the optimal switching. After offline training of the critic, which is a function of system states and elapsed time, the resulting optimal weights are used in online control, to get a smooth output AC voltage in a feedback form. Simulations show the desirable performance of this controller with linear and nonlinear load and its relative robustness to parameter uncertainty and disturbances. Furthermore, the proposed controller is upgraded so that the inverter is suitable for single phase variable frequency drives. Finally, as one of the few studies in the field of adaptive dynamic programming (ADP), the proposed controllers are implemented on a physical prototype to show the performance in practice.

2.1. Introduction

Uninterruptible power supply (UPS) units are generally used to supply high quality output with desired amplitude and frequency to critical loads such as life support systems, data storage units and telecommunication systems. Control of UPS must be capable to ensure that the output voltage has a low total harmonic distortion (THD) with negligible steady state error. Also it is crucial that the inverter be stable under both linear and nonlinear loads. Finally, since the system parameters usually change as the system ages, relative robustness to parameter uncertainties is a key factor in designing the inverter controller, [28, 43, 71].

One of the most common inverter topologies is voltage source inverter (VSI). There are many control approaches for this inverter such as proportional-resonant (PR) control [34], dead beat control [18], and hysteresis control [56] in the literature. Also, there has been investigations on more

complicated designs to have a better quality output voltage such as sliding mode control [11, 43], repetitive control, [16], multi-loop control, [67, 82], *dq*-frame current control, [49] and adaptive control, [80]. Most of the control approaches in the literature are based on pulse width modulation (PWM) technique. In this technique, achieving low total harmonic distortion (THD) is dependent on the switching frequency. Therefore, a trade-off is made between switching frequency and THD. Note that it is desirable to get a certain THD with least possible switching frequency due to the losses associated with switching, [55].

Variable-switching-frequency approaches are also proposed in the literature. In [41], a finite control set model predictive control (FCS-MPC) is proposed in which, for a specific number of switching combinations, a cost function is optimized. The key idea in FCS-MPC stems from limited number of switching combinations. At each step, the cost function is evaluated for all the switching combinations and the switching corresponding to the lowest cost is applied. The drawback of this approach is the trade-off between calculation load and prediction horizon. In [89], hysteresis MPC is proposed. In this method, the reference inverter current is estimated by extrapolation techniques. One drawback of this approach is the sensitivity of extrapolations to parameters. [40].

One of the promising fields of control theory is optimal control [38, 47, 68]. This method has been used successfully in inverter circuits. In [19], an optimal control problem, with dwell-time constraints, is solved numerically and the switching commands are generated based on a PWM-based approach. Robustness to initial conditions or system parameters is not guaranteed in this method because of the open loop nature of the design. In [74], linear quadratic regulator (LQR) method is used to trigger the appropriate switching for the inverter. In this approach, particle swarm optimization is utilized to optimize the penalizing matrices. In [20], LQR is used to design the multiple resonant controller gains. However, due to trial and error in choosing the weights, the system performance is greatly effected by these weights.

Adaptive dynamic programming (ADP) is a powerful tool for solving optimal control problems, [31, 39]. ADP is chosen in this research for two main reasons. First, we want to design an optimal variable switching frequency approach, which has lower number of switching, and hence, lower switching losses compared to PWM-based approaches. Second, we aim to design a controller,

which is relatively robust to parameter uncertainties. In our method, the first step is approximating the value function (i.e. optimal cost-to-go) from current time to final time. The second step, which leads to an optimal control in a feedback form, is the direct result of Bellman principle of optimality, [38]. In ADP design, value function and optimal control are approximated by two neural networks called 'critic' and 'actor', respectively, [22]. For switched systems, such as voltage source inverters, motivated by [23], the optimal infinite-horizon problem is solved only by using the critic network. In order to find the optimal switching schedule in online control, the critic network is trained offline [23]. The main difference between this work and FCS-MPC is the fact that, we are actually solving an infinite horizon problem, however FCS-MPC solves a one-step ahead problem. If FCS-MPC aims for longer horizons, the calculation load will be higher, however as will be seen in following sections, the calculation load in our approach is very low.

A thorough research in the literature yielded two works of [79] and [24], which are close to this study. In [79], an ADP-based approach is proposed for single phase inverters. The output of the actor network is restricted between the negative and positive input DC voltage. Afterwards, the continuous optimal control is discretized to actual values of negative and positive DC voltage by a hard limiter. In our work, however, no actor network is needed and the optimal switching modes are determined directly utilizing only one function approximator (critic). Also in [24], ADP is used in DC buck converters, however in that study the reference to be followed is fixed and the controller is more straightforward than tracking a harmonic reference signal done in this research.

Single phase inverters are typically designed to produce a fixed frequency output, which is desired for a specific application. However, it is also desired to design an inverter which can be used as a single phase variable frequency drive. With this type of inverter, the user can choose the output frequency of the inverter. This kind of inverter can be used with single phase AC motors to change the speed of the motor, [76]. Therefore, a part of this study is dedicated to designing an inverter which can incorporate both variable frequency and variable amplitude. This kind of inverter can be used in speed control of single phase induction motors with V/f technique, [13].

Compared to [35], which reports earlier results of this research, there are multiple improvements and new results in this study. First, the proposed controller is compared with a PWM-based sliding

mode controller as well as the FCS-MPC controller in order to show the effectiveness of ADP. Secondly, as one of the few studies in ADP, the proposed controller is implemented on a physical prototype and desirable results, especially in dealing with parameter uncertainty, are obtained in practice. Finally, the proposed controller is upgraded in order to achieve the capability of incorporating variable frequency and amplitude. With this upgrade, the inverter can actually be used as a single phase variable frequency drive (VFD).

As for organization of this study, Section 2.2 gives a brief introduction to single phase VSI. Afterwards, the optimal problem is formulated in Section 2.3, followed by simulation results in Section 2.4. In section 2.5, the proposed controller is modified to incorporate variable frequency and amplitude. The DC-AC inverter with the proposed controller is implemented and the results are provided in Section 2.6, followed by concluding remarks in Section 2.7.

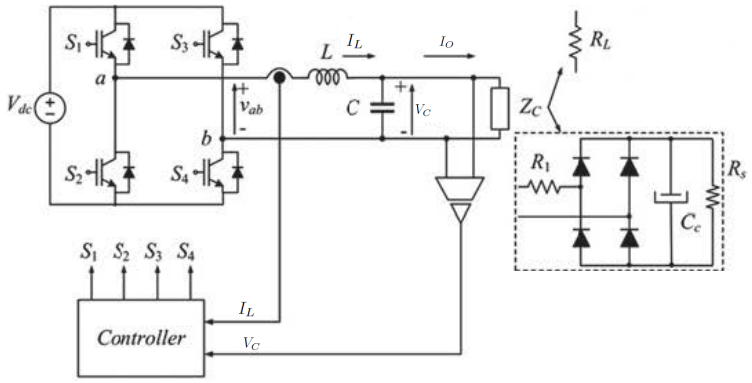


Figure 2.1: Single phase voltage source inverter circuit (picture adapted from [7]).

2.2. Single Phase Voltage Source Inverter

A single phase UPS inverter is shown in Fig. 2.1. It consists of a power stage, an output LC filter and the load, which is denoted by Z_C. The figure also shows the loads that are going to be considered in this research, including a nonlinear rectifier load. In order to find the state space representation of the circuit, we consider the load to be resistive R_L for now. The inductor current (I_L) and capacitor voltage (V_C) are chosen as the state variables, i.e., $x := [I_L, V_C]^T$. By this choice,

the state space realization becomes

$$\dot{x} = \frac{d}{dt}\begin{bmatrix} I_L \\ V_C \end{bmatrix} = \begin{bmatrix} L^{-1}(-r_L I_L - V_C + iV_{dc}) \\ C^{-1}(I_L - \frac{V_C}{R_L}) \end{bmatrix} = F_i(x), \tag{2.1}$$

where r_L is the internal resistance of the inductor. Table 2.1, shows different switching combinations of unipolar single phase VSI. It is seen that in 2.1, $i \in \{-1,0,1\}$, therefore the switched system has three modes.

By suitable switching of four switches $\{S_1,S_2,S_3,S_4\}$, we aim at inverting the input DC voltage to a desired output AC voltage. The desired output AC voltage is defined as $V_{out}(t) = V_m \sin(wt)$, where w is the angular frequency and t is time. The aim of the optimal controller is to determine the appropriate switching such that, the capacitor voltage (V_C) tracks the desired output voltage. Since the proposed approach is in discrete-time form, a proper sampling time is chosen to discretize the time and the forward Euler integration is used as

$$x_{k+1} = x_k + \Delta t F_i(x_k), \tag{2.2}$$

where x_k is the state at discrete time k.

2.3. Optimal Switching Using ADP

In this section, ADP is going to be used to solve the optimal control problem and find the optimal switching schedule. Let us define the switched system with M modes with the following nonlinear equation

$$x_{k+1} = f_i(x_k), k \in \mathbb{Z}_+, i \in \mathscr{I}, \tag{2.3}$$

where $x_k \in \mathbb{R}^n$ is the state vector at discrete time instances shown by non-negative integer k. Subscript i represents one of the M modes of the system and the system dynamics are given by

Table 2.1: Switch states of the unipolar single phase inverter

Switch State	S_1	S_2	S_3	S_4	V_{ab}
1	ON	OFF	OFF	ON	$+V_{dc}$
2	OFF	ON	ON	OFF	$-V_{dc}$
3	ON	OFF	ON	OFF	0
4	OFF	ON	OFF	ON	0

$f_i : \mathbb{R}^n \to \mathbb{R}^n, \forall i \in \mathscr{I} := \{1,2,...,M\}$. The discrete cost function is defined as:

$$J = \sum_{k=0}^{\infty} \gamma^k Q(x_k, V_{desk}) = \sum_{k=0}^{\infty} \gamma^k (V_{Ck} - V_{des_k})^2, \tag{2.4}$$

where $\gamma \in (0,1]$ is the discount factor which is used to ensure the boundedness of the cost. $Q : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}_+$ is a continuous, positive definite function, which penalizes the tracking error between actual capacitor voltage and desired voltage. V_{Ck} and V_{desk} denote the actual capacitor voltage and desired output voltage, respectively.

Since the cost function, and hence, the cost-to-go depends on states and time, therefore, the *optimal value function*, $V^* : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}_+$ is defined such that it takes the current states and current time as well. Therefore, states and time, at current instant, are fed to the optimal value function and the optimal cost is obtained, if optimal decisions are made from current instant onwards, [24]. Using the definition of optimal value function and considering V_{des_k} as a time function, (2.4) can be written as:

$$V^*(x_0, 0) := \sum_{k=0}^{\infty} \gamma^k Q(x_k^*, t_k), \tag{2.5}$$

where $x_0^* := x_0$ and x_k^*, $\forall k = 0,1,2,...$ is the optimal trajectory under optimal switching and t_k is the time at the kth step. This equation can be written recursively as:

$$V^*(x_k, t_k) := Q(x_k, t_k) + \gamma V^*(f_{i^*(x_k,t_k)}(x_k), t_{k+1}), \forall x_k \in \mathbb{R}^n, \forall k \in \mathbb{Z}_+, \tag{2.6}$$

where $i^*(x_k,t_k)$ is the *optimal* mode, $t_{k+1} = t_k + \Delta t$, and Δt is the sampling time. Using Bellman principle of optimality [38] we have

$$V^*(x_k,t_k) := \min_{i\in\mathscr{I}} \Big(Q(x_k,t_k) + \gamma V^*(f_i(x_k),t_{k+1}) \Big), \forall x_k \in \mathbb{R}^n, \forall k \in \mathbb{Z}_+, \tag{2.7}$$

and the optimal mode at each instant $i^*(x_k,t_k)$ is given by

$$i^*(x_k,t_k) := \arg\min_{i\in\mathscr{I}} \Big(Q(x_k,t_k) + \gamma V^*(f_i(x_k),t_{k+1}) \Big), \forall x_k \in \mathbb{R}^n, \forall k \in \mathbb{Z}_+, \tag{2.8}$$

Interested readers are referred to [46] for more complete derivation of (6) to (9). Therefore, in order to use (2.8), the value function at the next state and instant (x_{k+1},t_{k+1}) is evaluated and the optimal mode is the one that minimizes the corresponding value function. It is seen that optimal value function plays a crucial role in solving the optimal control problem. In order to learn the value function in a compact set, called *region of interest*, neural networks are utilized. The validity of approximation is guaranteed as long as the states and time remain in the region of interest. This closed set for state and time are denoted with Ω_x and Ω_t, respectively.

In order to solve (2.7), *Value iteration* is utilized. $V^0(x_k,t_k)$ is chosen as an initial guess for $V^*(x_k,t_k)$ then the following equation is used for updating $V^*(x_k,t_k)$,

$$V^{j+1}(x_k,t_k) := \min_{i\in\mathscr{I}} \Big(Q(x_k,t_k) + \gamma V^{j}(f_i(x_k),t_{k+1}) \Big), \forall x_k \in \Omega_x, \forall t_k \in \Omega_t, \tag{2.9}$$

where j is the iteration index. After convergence of the iterations, the optimal value function is obtained in the desired region of interest. According to [23], convergence of (2.9) to the optimal value function is guaranteed, if the initial guess is a continuous function such that $0 \leq V^0(x_k,t_k) \leq Q(x_k,t_k), \forall x_k \in \Omega_x, \forall t_k \in \Omega_t$. $V^0(x_k,t_k) = 0$ satisfies this condition and is chosen as the initial guess for starting the iterations. In this work, linear-in-parameter function approximator, is used to approximate the value function, which leads to

$$V(x_k,t_k) = W^T \Phi(x_k,t_k), \forall x_k \in \Omega_x, \forall t_k \in \Omega_t, \tag{2.10}$$

where $W \in \mathbb{R}^l$ is the critic weights matrix, $\Phi(x_k,t_k) \in \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^l$ is the set of smooth basis functions which are chosen by the user and l is the number of neurons. Therefore, we can implement value iteration scheme in (2.9) with the help of function approximators in (2.10). At each iteration a new set of weights are obtained and the iteration is continued until convergence of the weights. Substituting (2.10) in (2.9) we get

$$W^{j+1^T}\Phi(x_k,t_k) := \min_{i\in\mathscr{I}} \Big(Q(x_k,t_k) + \gamma W^{j^T}\Phi(f_i(x_k),t_{k+1})\Big), \forall x_k \in \Omega_x, \forall t_k \in \Omega_t. \tag{2.11}$$

Suitable number of random samples for x_ks and t_k are fed to (2.11) and least square is utilized for solving the equation for each W^j. Interested readers are referred to [27] for further discussion on finding weights with least square. After convergence of the weights, the final weight matrix W^* is used in (2.10) to approximate the optimal value function. After *offline* learning of W^*, *online* control is achieved in a feedback form by the following equation:

$$i^*(x_k,t_k) := \arg\min_{i\in\mathscr{I}} W^{*T}\Phi(f_i(x_k),t_{k+1}), \forall x_k \in \Omega_x, \forall t_k \in \Omega_t. \tag{2.12}$$

Equation (2.12) states that in order to find the optimal switching at each instant initially, the next state, which is found by the model for each switching mode, and the next time instant, which is $t_k + \Delta t$ are fed to the neural network. Then the optimal switching schedule is the one that minimizes $W^{*T}\Phi(f_i(x_k),t_{k+1})$, for all system modes. The reason for dropping $Q(x_k,t_k)$ in (2.12) is its independency to the modes (i).

2.4. Implementation on Voltage Source Inverter

In this section, the proposed controller based on ADP is implemented on the single phase VSI and the performance of this controller with different loads is investigated. Also relative robustness of the controller is shown by some simulations. In order to do a comparison with a PWM-based method, the circuit parameters, given in Table 2.3, are chosen from [43].

Table 2.3: Parameters of the single phase inverter shown in Fig. 2.1 (adapted from [43].)

Parameter	Value
Input Voltage, V_{dc}	$275\ V$
Output Voltage, V_{ac}	$120\sqrt{2}V$
Base Frequency, f_r	$50\ Hz$
Sampling Time, T_s	$16\ \mu s$
Output Filter Inductance, L	$250\ \mu H$
Inductor Resistance, r_L	$0.2\ \Omega$
Output Filter Capacitance, C	$100\ \mu F$
Rated Resistance, R_L	$30\ \Omega$
Non-linear Reference Load Resistance, R_s	$80\ \Omega$
Non-linear Reference Smoothing Resistance, R_1	$0.5\ \Omega$
Non-linear Reference Load Capacitance, C_c	$400\ \mu F$

2.4.1. Neural Network Training

As discussed in Section 2.3, the region of interest for states and time (Ω_x and Ω_t) must be selected to do the training. Using similar or close range for the data usually leads to better approximations in the training stage of neural networks, [42]. As seen in Table 2.3, $V_{Cmax} = 120\sqrt{2}V$ and some trial and error lead to $I_{Lmax} = 20A$. Moreover, the period of the desired harmonic voltage (i.e. $0.02s$), is a good choice for the range of change of *time*. The following normalization is used to bring states and time to a similar region,

$$I_L = a\tilde{I}_L, V_C = b\tilde{V}_C, t = \tau\tilde{t}, \tag{2.13}$$

where $a = 20$, $b = 120\sqrt{2}$ and $\tau = 0.02$, and the normalized quantities are denoted with the added '˜' notation. The following equation is written based on the normalized states and time:

$$\dot{\tilde{x}} = \frac{d}{d\tilde{t}}\begin{bmatrix}\tilde{I}_L \\ \tilde{V}_C\end{bmatrix} = \tau\begin{bmatrix}L^{-1}(-r_L\tilde{I}_L - \frac{b}{a}\tilde{V}_C + i\frac{V_{dc}}{a}) \\ C^{-1}(\frac{a}{b}\tilde{I}_L - \frac{\tilde{V}_C}{R_L})\end{bmatrix} = f_i(\tilde{x}). \tag{2.14}$$

Based on (2.13), regions of interest for training are chosen as $\Omega_x = \{\begin{bmatrix}\tilde{I}_L, \tilde{V}_C\end{bmatrix}^T \in \mathbb{R}^2 : -1.5 \leq \tilde{I}_L, \tilde{V}_C \leq 1.5\}$ and $\Omega_t = \{\tilde{t} \in \mathbb{R} : 0 \leq \tilde{t} \leq 1.5\}$. In order to have better generalization near the boundaries, the training region is selected such that the maximum value of the normalized states, which is 1, is inside the region of interest. The basis functions are chosen from the set $\{\cos(r_1 wt){x_1}^{r_2}{x_2}^{r_3}, \sin(r_4 wt){x_1}^{r_2}{x_2}^{r_3}\}$, where non-negative integers r_is, $i = 1,...,4$ are selected as $r_1 = \{0,1,2,3\}$, $0 \leq r_2 + r_3 \leq 4$, and $r_4 = \{1,2,3\}$ which leads to 105 neurons. Note that these basis functions are chosen based on the Weierstras approximation theorem and the fact that the desired reference is a trigonometric function. Nevertheless, the user may choose other types of basis functions as these are design parameters and choices. In order to use batch learning algorithm [23], 4000 random states and time for training are picked. Using least square method, (2.11) is iterated for $\gamma = 0.3$. Fig. 2.2 shows the weights converging after 8 iterations which approximately took 9 seconds on a desktop computer with Intel Core i-5-6500, 3.2GHz processor and 16GB of RAM, running MATLAB 2018a. The history of weights, including the final weights, can be found in [3].

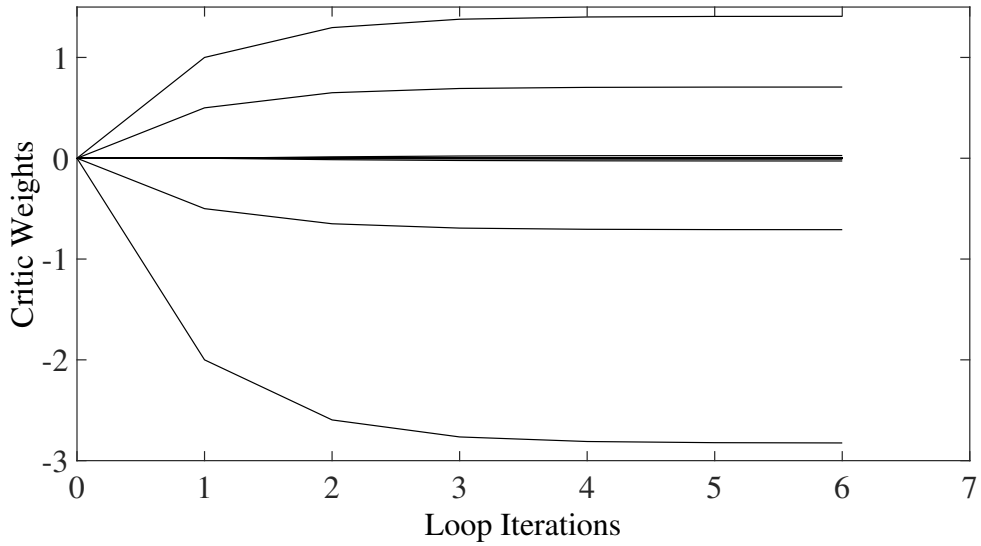


Figure 2.2: History of weights during learning iterations for fixed amplitude and frequency

2.4.2. Performance With Different Loads

A linear resistive load $R_L = 30\Omega$ is chosen for the first simulation. The obtained converged weights are used in (2.12) and the simulation is done for initial condition of $x_0 = [0,0]^T$. The output voltage and output current are shown in Fig. 2.3. The THD of the voltage is 0.7% and the maximum switching frequency is $11.1kHz$, however for the same THD, the switching frequency of [43] is $13.42kHz$. Note that if $13.42kHz$ is chosen as the maximum switching frequency for ADP-based controller, the THD will reduce significantly to only 0.4%. Fig. 2.4 shows the active modes. Parts of the figure are magnified to verify the variable switching frequency nature of the method.

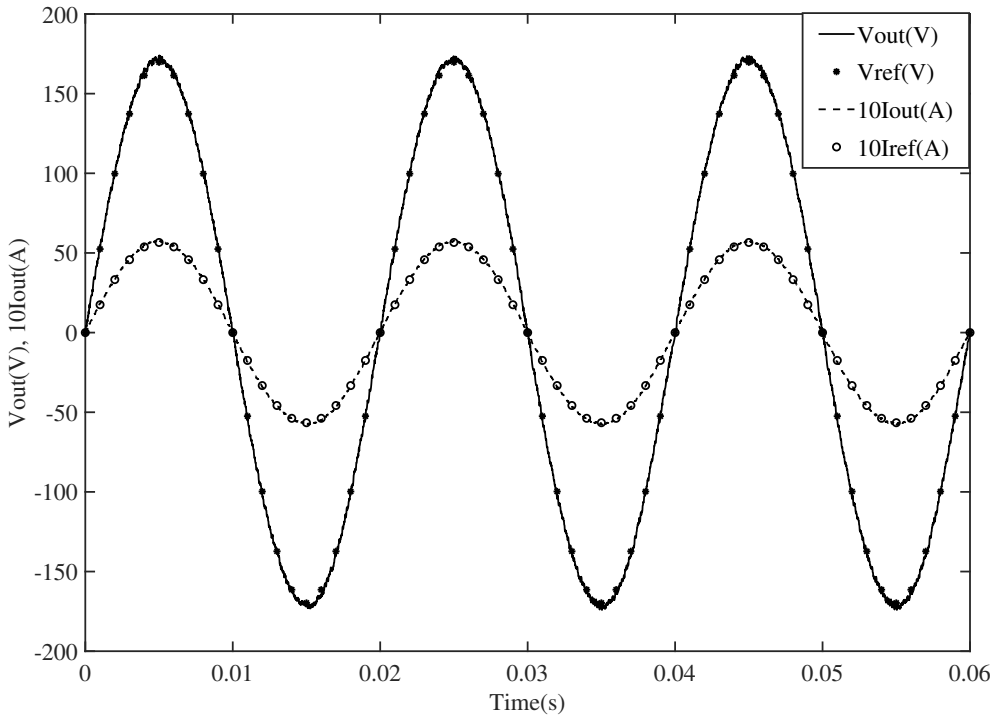


Figure 2.3: Output voltage and output current for a linear resistive load

If the states and time continue to be inside the region of interest, the controller is reliable. Fig. 2.5.a shows the output voltage of the proposed controller for different initial states, which include $x_0 = [0A, 0V]^T$, $x_0 = [0A, -20V]^T$ and $x_0 = [0.183A, 20V]^T$ when the load is a linear resistance as before. Satisfactory response is obtained after a fast transient. Note that in order to ensure that the *time* remains in the desired region of interest, the remainder of it, devided by the period, i.e.,

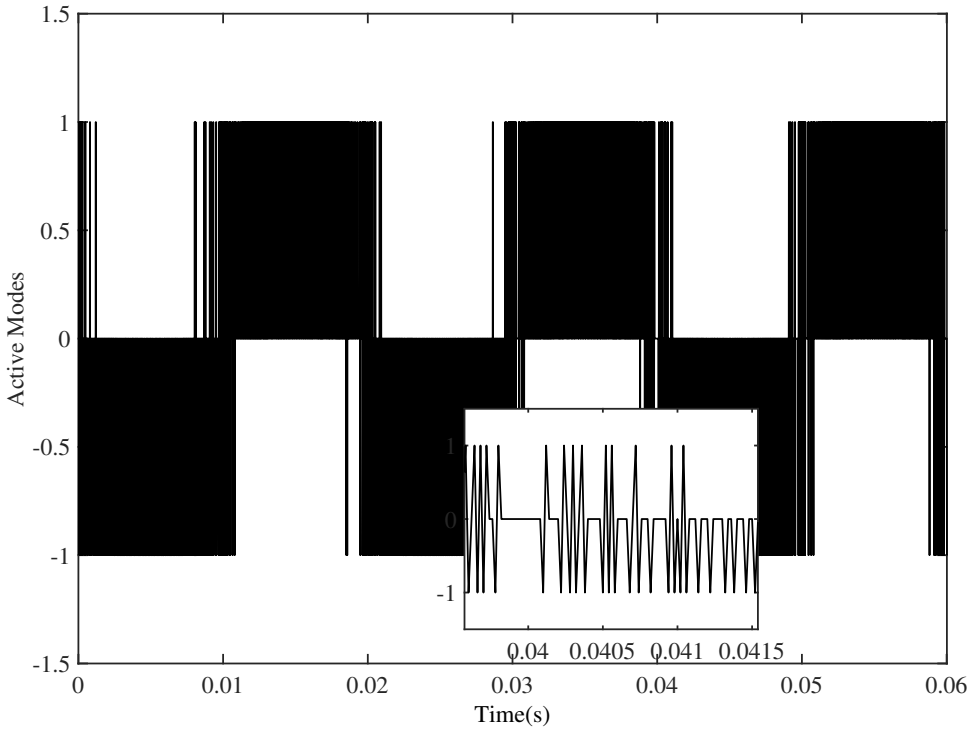


Figure 2.4: Active modes for a linear resistive load

τ/f_r is used. The reasoning behind this is that if everything else is fixed, the switching policy is periodic in time, with the period τ/f_r. As mentioned in section 2.1, the FCS-MPC only considers one step ahead, while the proposed method takes into account the infinite horizon. In order to do a comparison, the same initial conditions are chosen for simulation of FCS-MPC with the results given in 2.5.b. Comparison of the proposed scheme (Fig. 5.a) with FCS-MPC (Fig. 5.b) shows the better performance of the ADP-based scheme.

A nonlinear rectifier load, based on EN-62040 Standard as a typical load for UPS systems, [69], is chosen for the second case. Fig. 2.1 shows this load with parameter values given in Table 2.3. Because of its nonlinearity, this load typically causes harmonic distortions in the output voltage [1]. Fig. 2.6 shows the output voltage, output current and the error between desired reference and obtained output. For the same switching frequency as [43], i.e. $13.42kHz$, The THD of the output voltage is calculated to be 0.9%, therefore the IEEE standard 1547 is easily met, for nonlinear rectifier load (i.e.,THD$\leq 5\%$) [1]. Note that if we decide to have the same THD of [43], which is

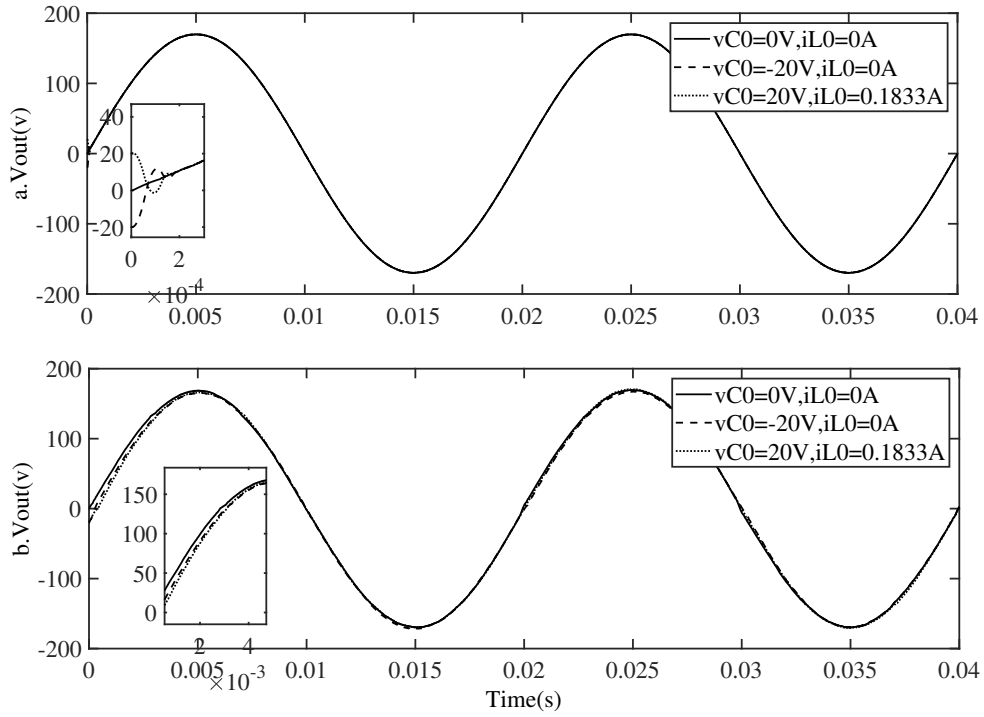


Figure 2.5: Output voltage of linear resistive load with different initial conditions for ADP (Fig. 5.a) and FCS-MPC (Fig. 5.b)

1.51%, the maximum switching frequency of ADP-based controller will be 11.5kHz.

Note that, the system is trained for a resistive load. But, in online control when nonlinear load is considered, the system dynamics change. However, the proposed controller has been effective in handling these variations, thanks to its feedback nature. In other words, the feedback controller has rejected the 'disturbance' on the system, which is caused by the change in the load. Finally, it may be noted that this research has not studied the effect of disturbance on the controller design theoretically. Therefore such a performance is not guaranteed for significantly more considerable and intense disturbances.

2.4.3. Sensitivity to System Parameters

In order to see the effect of parameter uncertainties on performance of the controller, three scenarios are considered in this section. In both scenarios, the nominal values of parameter are used

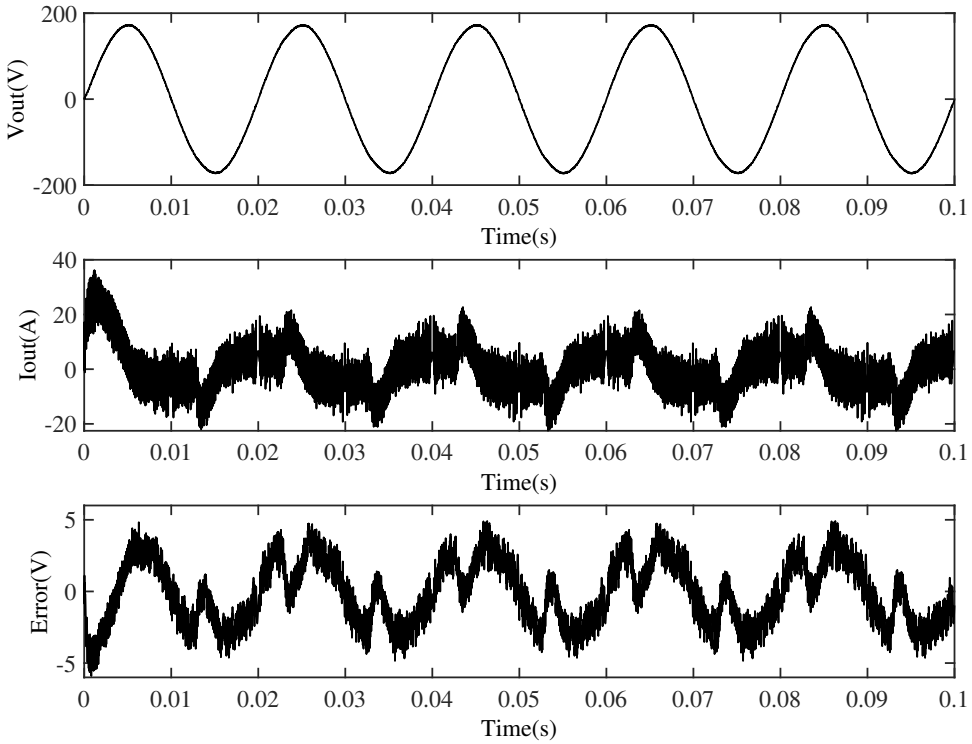


Figure 2.6: Output voltage, current and voltage error of nonlinear load

in offline stage to train the network, however in online control, there will be changes in system parameters and simulations will be done 'without re-tuning' the controller. Hence, the controller is 'kept in the dark' for the purpose of evaluating its capability in handling such uncertainties.

Initially a step change is applied in the load from no load to $R_L = 30\Omega$ and the effect is analyzed. As seen in Fig. 2.7, at $t = 0.025s$ the load has increased from zero to 30Ω. It is observed that the system tracks the reference voltage properly, despite the sudden change in the load. The second scenario considers the case when there is a 30% mismatch between system parameters in training and online control. For this case, training is performed with $L = 250\mu H$, $r_L = 0.5\Omega$, and $C = 100\mu F$ as given in Table 2.3, however, online control is obtained with $L = 175\mu H$, $r_L = 0.35\Omega$, and $C = 70\mu F$. Fig. 2.8 shows the desirable tracking with THD of 1.2%. Therefore, relative robustness to parameter and load uncertainty has been demonstrated by these simulations. Note that the switching policy is obtained using the parameter values used in the training (i.e., $L = 250\mu H$,

$r_L = 0.5\Omega$, and $C = 100\mu F$). However, the states are propagated with the actual values of the parameters (i.e. $L = 175\mu H$, $r_L = 0.35\Omega$, and $C = 70\mu F$).

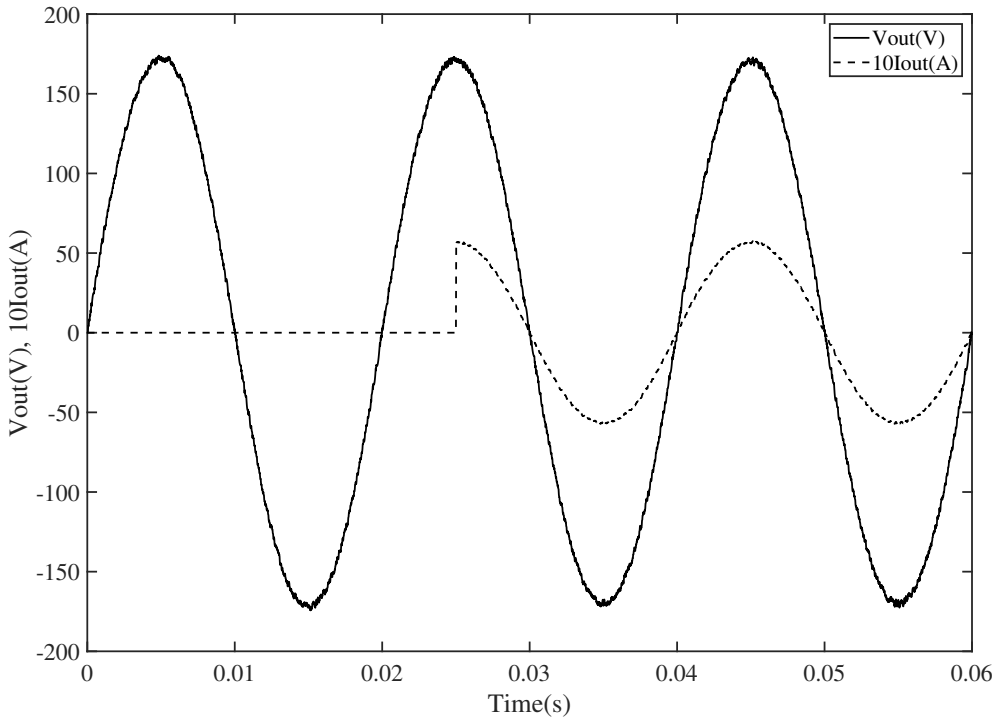


Figure 2.7: Output voltage and current for linear resistive load with a step change from no load to $R_L = 30\Omega$

As the final numerical analysis on sensitivity, let us consider the effect of change of the input DC voltage on the output. Suppose the input DC voltage changes as follows

$$\begin{cases} V_{DC} = 275V & \text{if } 0s \leq t \leq 0.045s \\ V_{DC} = 220V & \text{if } 0.045s \leq t \leq 0.1s \\ V_{DC} = 320V & \text{if } 0.1s \leq t \leq 0.2s \end{cases} \tag{2.15}$$

The disturbance in line voltage should be rejected by the inverter. The output waveform with the new input DC voltage is shown in Fig. 2.9.a, which demonstrates a desirable tracking of the reference. In order to observe the superior performance of the proposed feedback controller, the switching policy for the case of constant input voltage of 300V is applied to the system under changing input DC

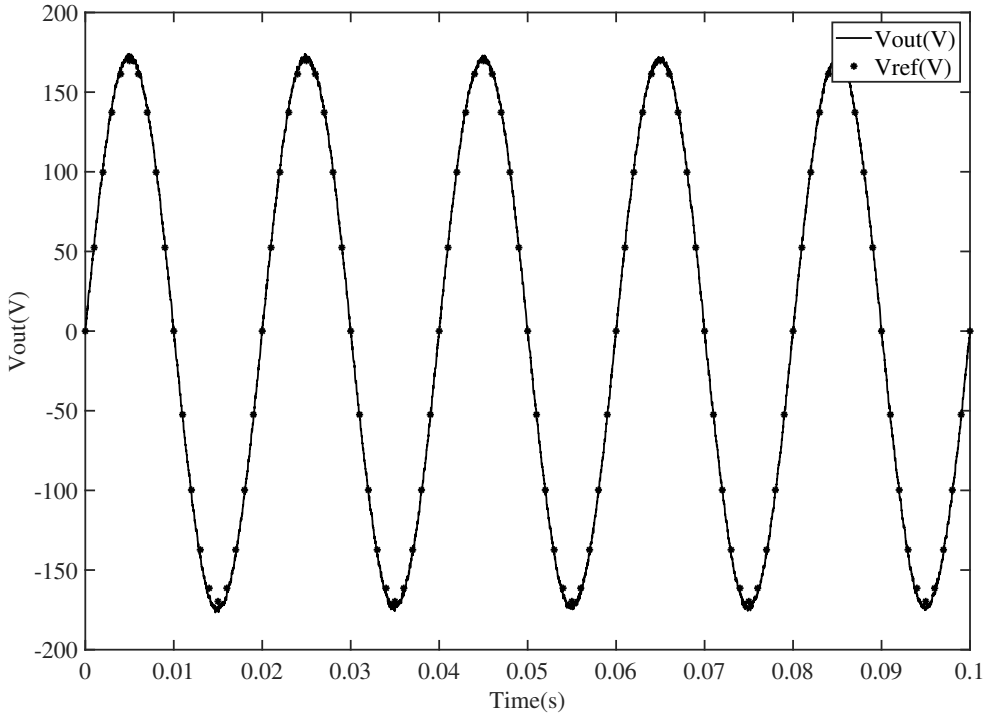


Figure 2.8: Output voltage of linear resistive load when there is a mismatch in system parameters

voltage. Let us call this case "open-loop controller". Fig. 2.9.b shows the response of the open-loop controller under DC voltage disturbances. It is seen that without feedback, the output voltage can not track the desired reference correctly.

Finally, it should be noted that this research does not claim that the controller is robust to structured or unstructured uncertainty. However, because of feedback nature of the controller, it is seen in numerical examples that uncertainties and disturbances are managed to some extent. As stated before, a desired performance under more significant uncertainties is not guaranteed.

2.5. Single Phase Variable Frequency Drive

Up to this point, the proposed inverter is "single purpose", in the sense that it can invert the input DC voltage to an AC voltage with a fixed frequency and amplitude. In this section, it is shown that with some modifications on the critic, the controller is capable to incorporate variable frequency and amplitude. Therefore, it can be a "multi purpose" inverter in which, the user can define the

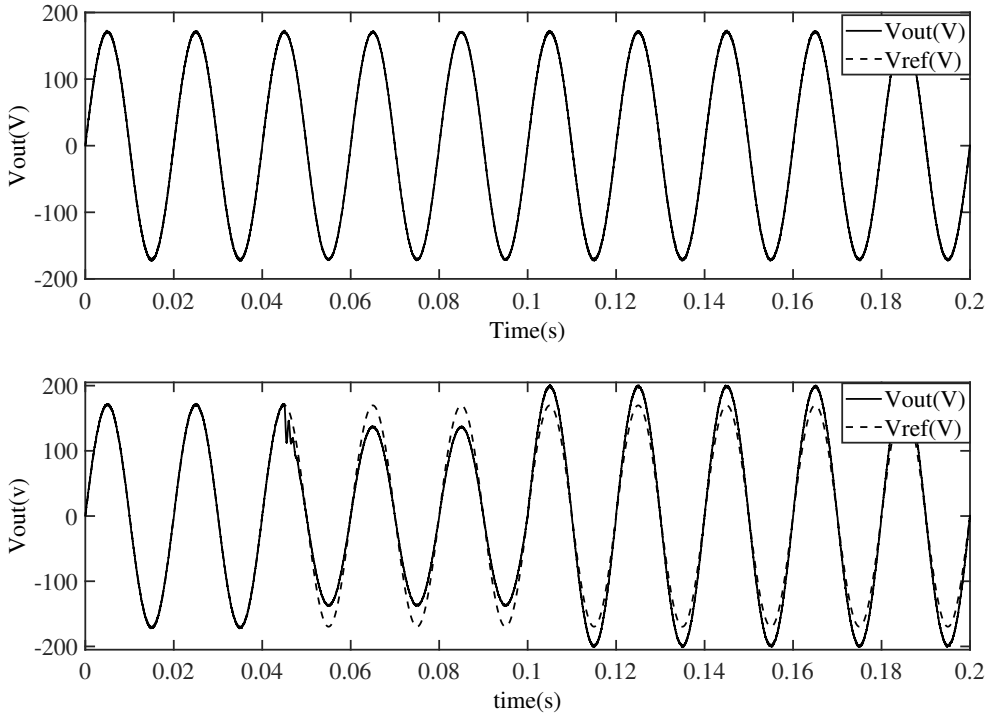


Figure 2.9: A. Output voltage of the feedback controller under changing input DC voltage B. Output voltage of the open-Loop controller under changing input DC voltage

frequency and amplitude of the output voltage on the fly. This kind of inverter can be used in single phase variable frequency drives (VFD).

In order to see the performance of the previously designed controller with variable frequency and amplitude, let us first consider a scenario in which, the amplitude and frequency of the desired reference voltage $V_{out}(t) = V_m \sin(wt)$ changes as follows.

$$\begin{cases} V_m = 120\sqrt{2}, f_r = 50Hz & \text{if } 0s \leq t \leq 0.05s \\ V_m = 0.8 \times 120\sqrt{2}, f_r = 50Hz & \text{if } 0.05s \leq t \leq 0.1s \\ V_m = 1.2 \times 120\sqrt{2}, f_r = 60Hz & \text{if } 0.1s \leq t \leq 0.15s \\ V_m = 120\sqrt{2}, f_r = 35Hz & \text{if } 0.15s \leq t \leq 0.2s \end{cases} \tag{2.16}$$

Fig. 2.10 shows the output voltage for the case when the previously trained critic network is used to generate the switching policy. As seen in the figure, the output voltage can not track the reference. This performance is not surprising because the network is trained for a specified frequency $f_r = 50Hz$ and amplitude $V_m = 120\sqrt{2}$.

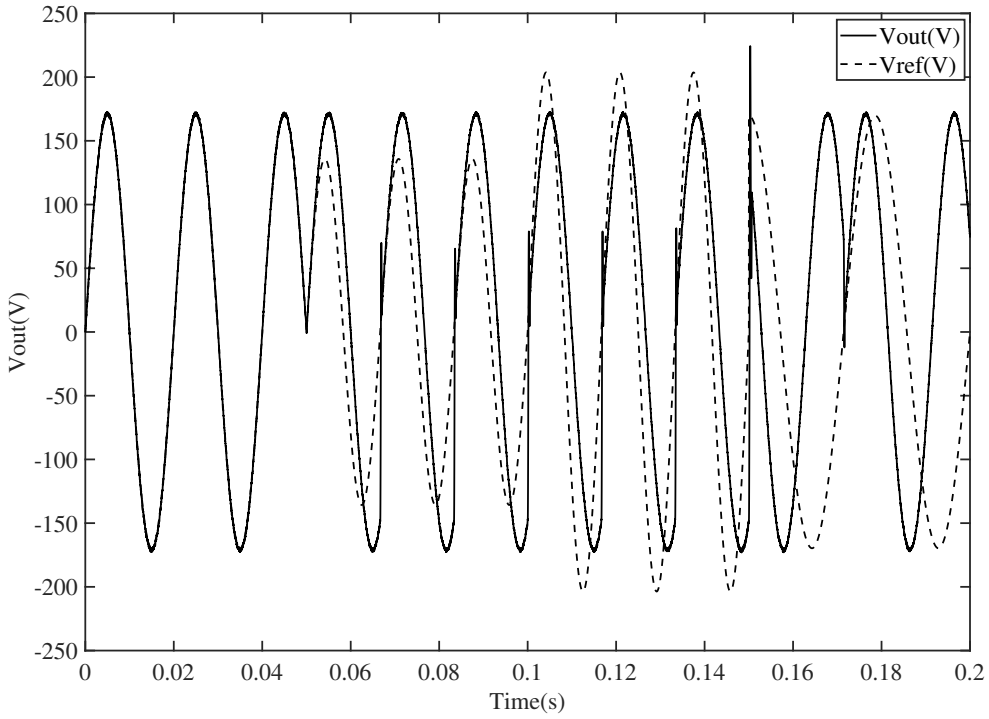


Figure 2.10: Output voltage for variable frequency and amplitude when only states and time are fed to the critic network

Motivated by [24] in a different context and for a different concern, an approach is proposed for solving this problem. Since in this case, amplitude and frequency are variable, these parameters also are parts of the set of parameters on this the value function depends. Therefore, they also should be fed to the critic network. Therefore, the optimal value function $V^* : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ and cost function term $Q : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ are not only dependent on states and time, but also on amplitude, denoted by V_m and angular frequency, denoted by ω. Therefore, (2.7) and (2.8) are

modified as follows:

$$V^*(x_k,t_k,w_k,V_{mk}) := \min_{i\in\mathscr{I}} \Big(Q(x_k,t_k,w_k,V_{mk}) + \gamma V^*(f_{i^*(x_k,t_k,w_k,V_{mk})}(x_k),t_{k+1},w_k,V_{mk}) \Big) \tag{2.17}$$

$$i^*(x_k,t_k,w_k,V_{mk}) := \arg\min_{i\in\mathscr{I}} \Big(Q(x_k,t_k,w_k,V_{mk}) + \gamma V^*(f_i(x_k),t_{k+1},w_k,V_{mk}) \Big) \tag{2.18}$$

Both equations should hold for $\forall x_k \in \mathbb{R}^n, \forall w_k \in \mathbb{R}, \forall V_{mk} \in \mathbb{R}, \forall k \in \mathbb{Z}_+$. It should be noted that the logic behind (2.17) is the fact that all the states, time, amplitude and frequency values, when chosen from their respective region of interest, should satisfy this equation.

The basis functions are chosen from the set $\{V_m \cos(r_1 wt) {x_1}^{r_2} {x_2}^{r_3}, V_m \sin(r_4 wt) {x_1}^{r_2} {x_2}^{r_3}\}$ with non-negative integers $r_1 = \{0,1,2,3\}$, $0 \le r_2 + r_3 \le 5$, and $r_4 = \{1,2,3\}$ are selected. This selection leads to 147 neurons. Based on the normalized states and time as before, and the range of changes for frequency and amplitude, regions of interest for training are chosen as $\Omega_x = \{ \begin{bmatrix} \tilde{I}_L, \tilde{V}_C \end{bmatrix}^T \in \mathbb{R}^2 : -1.5 < \tilde{I}_L, \tilde{V}_C < 1.5\}$, $\Omega_t = \{\tilde{t} \in \mathbb{R} : 0 < \tilde{t} < 1.5\}$,$\Omega_w = \{\tilde{w} \in \mathbb{R} : 0 < \tilde{w} < 4\pi\}$, and $\Omega_{V_m} = \{V_m \in \mathbb{R} : 0 < V_m < 1.5\}$. Using value iteration to solve (2.17), the iteration given by (2.11) is replaced with

$$\begin{aligned} {W_{critic}^{j+1}}^T \Phi(x_k,t_k,w_k,V_{mk}) &:= \min_{i\in\mathscr{I}} \Big(Q(x_k,t_k,w_k,V_{mk}) + \gamma {W_{critic}^{j}}^T \Phi(f_i(x_k),t_{k+1},w_k,V_{mk}) \Big) \\ &\forall x_k \in \Omega_x, \forall t_k \in \Omega_t, \forall w_k \in \Omega_w, \forall V_{mk} \in \Omega_{V_m} \end{aligned} \tag{2.19}$$

The number of 100000 random states, time, amplitude and frequency points are chosen for training and the iterative learning given by (2.19) is used with $\gamma = 0.5$ using least square. Note that since value function is dependent on *four* parameters, more points are needed for accurate training. Fig. 2.11 shows the converged weights for this case. The history of weights, including the final weights, can be found in [3]. It takes 12 iterations which lasts almost 4 minutes to do the training with the same desktop computer. Although it takes a relatively longer time for training, there is

no need to retrain the system again for other parameters, as long as the parameters remain in their respective regions used in training. After the offline training, we can use the following equation in online control:

$$
\begin{aligned}
& i^*(x_k, t_k, w_k, V_{mk}) := \arg\min_{i \in \mathscr{I}} W^{*T}_{critic} \Phi(f_i(x_k), t_{k+1}, w_k, V_{mk}) \\
& \forall x_k \in \Omega_x, \forall t_k \in \Omega_t, \forall w_k \in \Omega_w, \forall V_{mk} \in \Omega_{V_m}
\end{aligned}
\tag{2.20}
$$

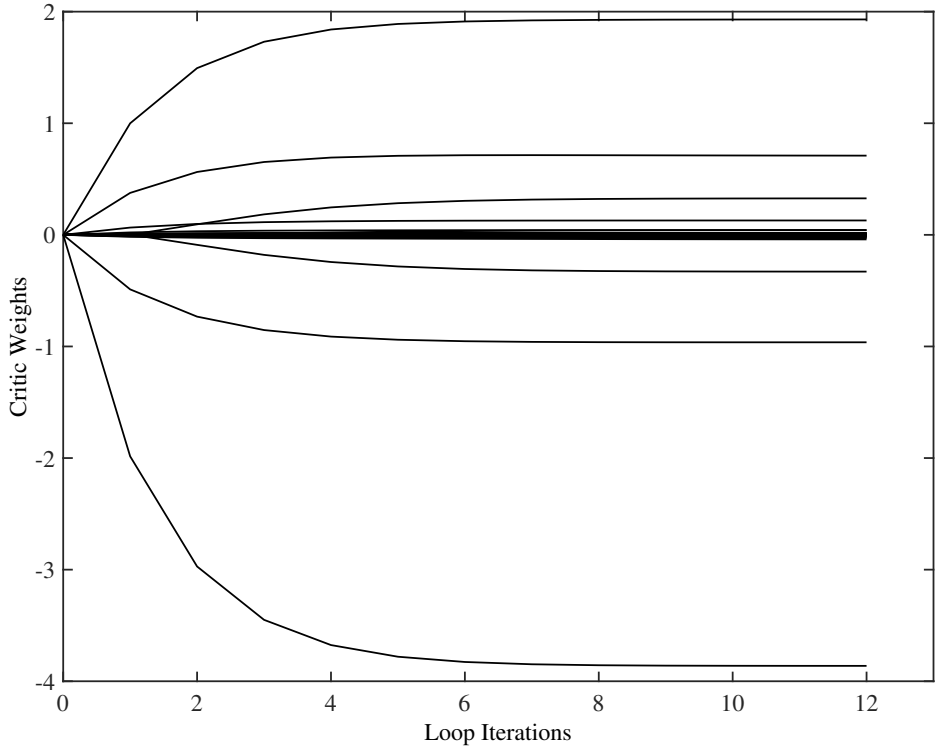


Figure 2.11: History of weights during learning iterations for variable amplitude, frequency, and phase angle

This equation is the same as (2.12), except it takes into account the desired frequency and amplitude. Therefore, the same logic behind (2.12) can be extended to (2.20). Using the same scenario as (2.16), the output voltage is given in Fig. 2.12. It is observed that the tracking is desirable and the performance is significantly better than before, i.e., compared with Fig. 2.10.

Also, the sharp point at $t = 0.05s$ is magnified to show the fast transition of the response.

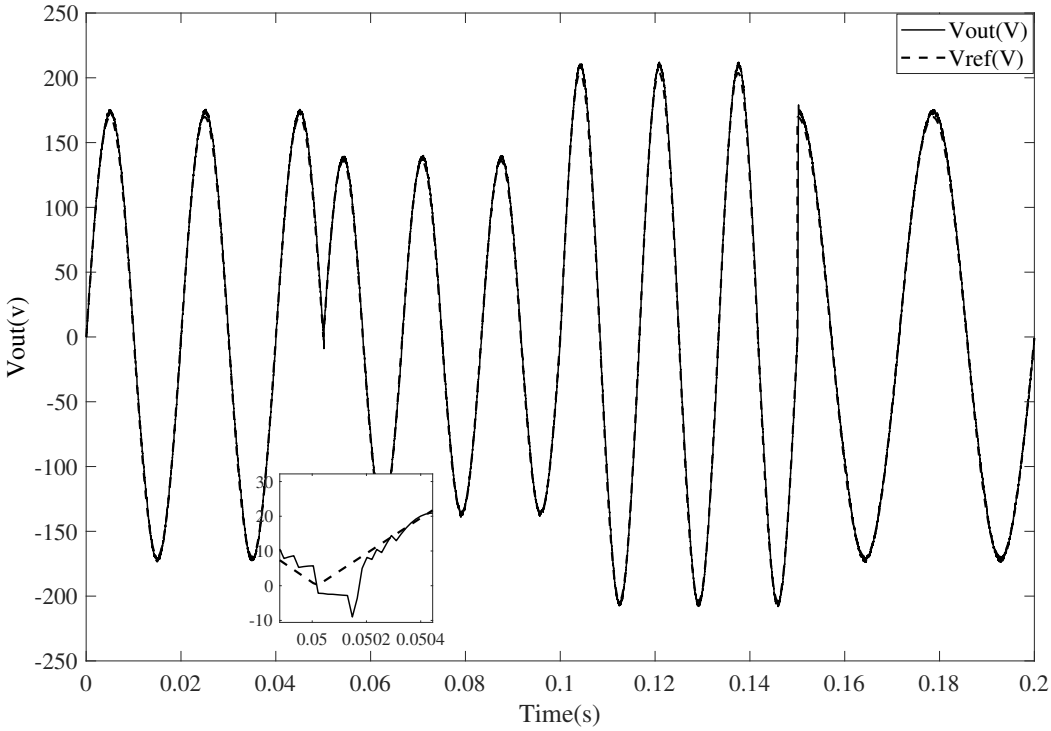


Figure 2.12: Output voltage for variable frequency and amplitude when states, time, amplitude, frequency and phase angle are fed to the critic network

This result is interesting since the system is trained once for a range of change of the four parameters, including amplitude and frequency and then it is used in control. In other words, there is no need for retraining, to incorporate the changes in the parameters. Therefore, After the training is done, in online stage, the user can change the frequency and amplitude to get the desired output voltage (as long as the parameters remain in their respective regions used in training).

2.6. Experimental Results

In order to show the effectiveness of the proposed method, a single phase inverter is built and tested. Due to safety issues in the lab environment, a low power inverter is designed for the experiment. The experimental platform consists of a full bridge MOSFET, gate drivers, op-amp circuits for measuring voltage and current, and an LC filter. The proposed controller is implemented

on a TMS320F28379D floating point digital signal controller from Texas Instrument.Table 2.5 shows the parameters of the implemented circuit and the schematic figure can be found in [3].

Fig. 2.13 shows the implemented circuit on a PCB connected to the micro-controller. In order to do a fair comparison, a multi-loop PWM-based controller proposed in [60], which is relatively straightforward to implement is chosen and the same hardware is utilized for its implementation.

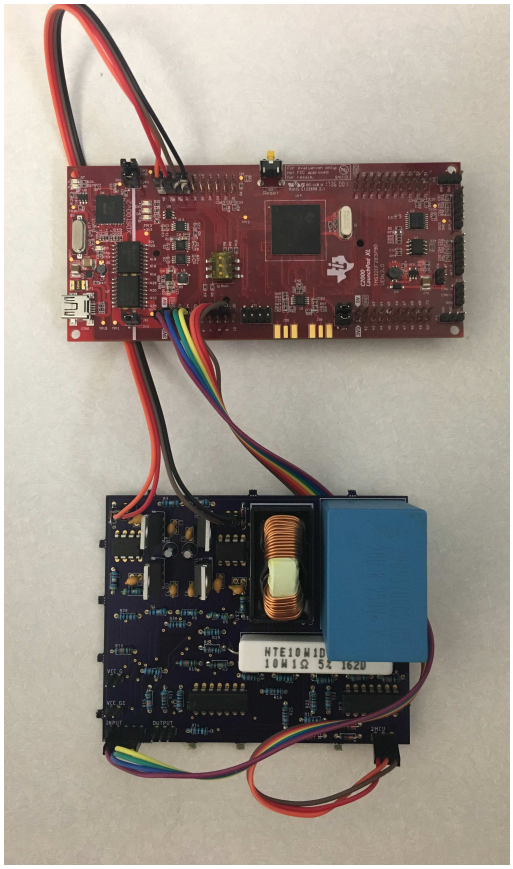


Figure 2.13: Implemented circuit of the single phase inverter

The capacitor voltage and inductor current are measured by the help of op-amps and then fed to the analog-to-digital converter channels of the digital signal controller. Moreover, the elapsed time is measured by utilizing the CPU timer of the digital signal controller. In other words, the CPU timer measures the time difference between current instant and the next instant. Also in order to prevent overflow, the CPU timer in reset at the start of each instant. This elapsed time is used to calculate t_{k+1} in (2.12). In online control, the computational load of the switching policy is calculating three scalar values and then comparing them. These scalar values are computed using

the pre-calculated optimal weight, W^*_{critic} and basis functions $\Phi(x,t)$. The basis functions are chosen from the set $\{\cos(r_1 wt){x_1}^{r_2}{x_2}^{r_3}, \sin(r_4 wt){x_1}^{r_2}{x_2}^{r_3}\}$, where non-negative integers r_is, $i = 1,...,3$ are selected as $r_1 = \{0,1,2\}$, $0 \leq r_2 + r_3 \leq 2$, and $r_4 = 1$, which leads to 19 neurons.

Table 2.5: Parameters of the implemented single phase VSI shown in Fig. 2.1

Parameter	Value
Input Voltage, V_{dc}	10 V
Output Voltage, $V_{ac}(max)$	5 V
Fundamental Frequency, f_r	50 Hz
Sampling Time, T_s	20 μs
Output Filter Inductance, L	5 mH
Inductor Resistance, r_L	0.02 Ω
Output Filter Capacitance, C	20 μF
Rated Resistance, R_L	100 Ω

Fig. 2.14 shows the output voltage of the single phase voltage source inverter for a resistive load. It is seen that the output voltage is relatively smooth with frequency of 50Hz and the amplitude of 5V as desired. Using the FFT function of the oscilloscope, the THD is calculated to be 0.3%, which is a desired result.

In order to test the performance of the controller under nonlinear load, the rectifier load with $R_1 = 4.3\Omega$, $R_s = 1k\Omega$ and $C_c = 20\mu F$ is chosen. Fig. 2.15 and Fig.2.16 show the output voltage and output current of the nonlinear load, respectively. The THD of the output voltage is 0.35%.

Fig. 2.17 shows the no load to full load transition. A 100Ω resistor is used as the full load. It is seen that the transition is pretty smooth and the output is not distorted when it goes under full load.

In order to show the effectiveness of the proposed controller in dealing with parameter uncertainties, during online control, the filter capacitor is changed to $4.5\mu F$. Note that the scheduler is still using the $20\mu F$ value to find the switching schedule. Fig. 2.18 shows the output voltage under this parameter mismatch. It is seen that although there is a big difference between the trained parameter and actual parameter during online stage, the output voltage is acceptable.

The same scenario of parameter mismatch is applied on a multi-loop PWM-based approach, proposed in [60]. In order to have a fair comparison, the same components are used for implementation of this controller. Fig. 2.19 shows the output voltage and it is seen that under the same conditions, the ADP controller results in a much better output voltage. It is important to note that the comparison is made with a special class of PWM-based inverters (i.e. multi-loop inverter), and certainly it does not mean that all PWM-based methods would result in an undesirable performance with this scenario.

As mentioned in section 2.5, with some modifications in the critic network, it is possible to upgrade the proposed inverter to the one which can be used as a single phase VFD .Next this capability is evaluated experimentally. The basis functions are chosen from the set $\{V_m \cos(r_1 wt) {x_1}^{r_2} {x_2}^{r_3}$, $V_m \sin(r_4 wt) {x_1}^{r_2} {x_2}^{r_3}\}$ with non-negative integers $r_1 = \{0,1,2\}$, $0 \leq r_2 + r_3 \leq 2$, and $r_4 = \{1\}$. This selection leads to 24 neurons. After training the new network, the converged weights are used in the experiment. Fig. 2.20 shows an experiment in which the user has changed both the frequency and amplitude. It is observed that after a very fast transient period, the output voltage reaches the desired reference with specified frequency and amplitude.

2.7. Conclusion

A non-PWM variable-switching-frequency method based on adaptive dynamic programming is proposed to convert DC voltage to AC voltage. The critic network is trained once offline and the converged weights are used in online control to determine the optimal switching schedule. The performance of the controller under different loads is analysed both with simulations and experiments. Finally, the controller is modified and the inverter is upgraded to be used as single phase VFD and the results are shown both numerically and experimentally.

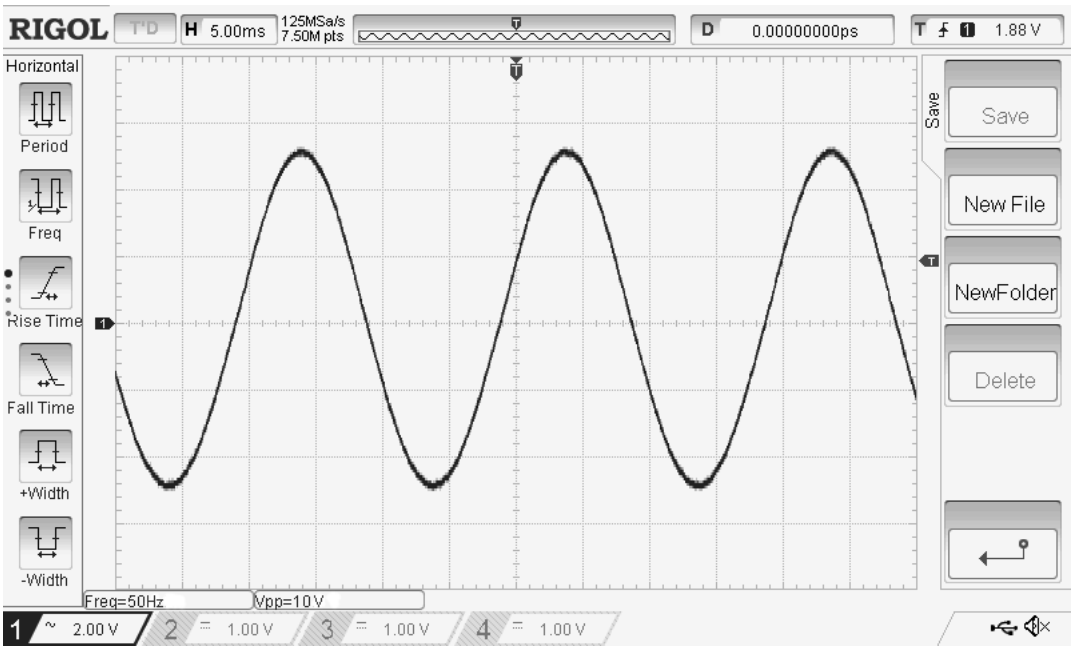


Figure 2.14: Experimental result of the output voltage for a linear resistive load with THD=0.3%

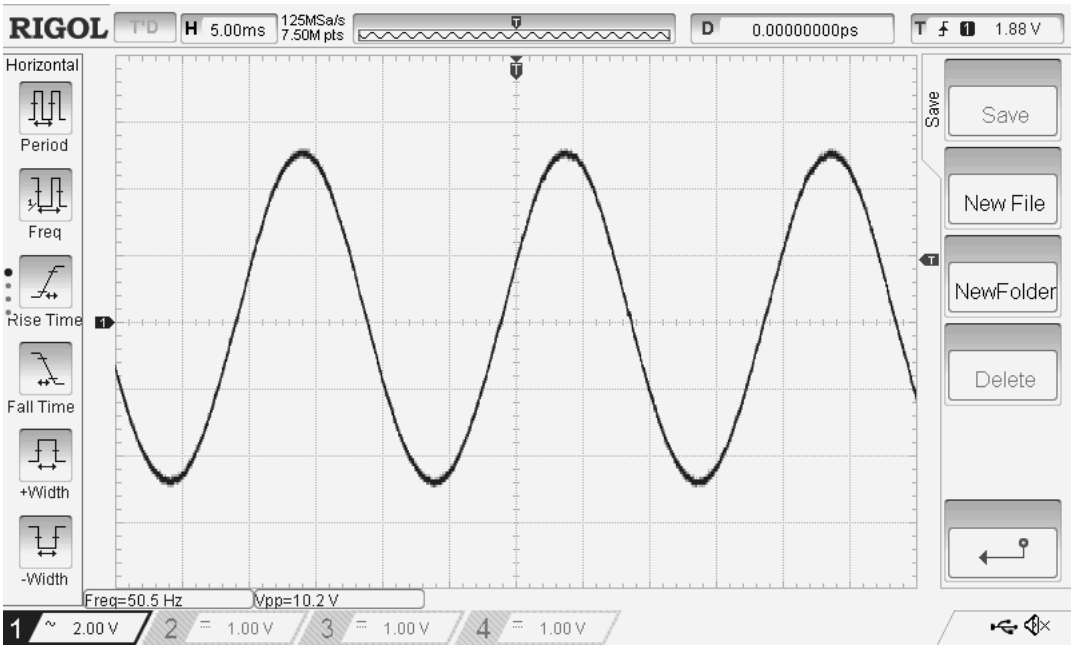


Figure 2.15: Experimental result of the output voltage for a nonlinear load with THD=0.35%

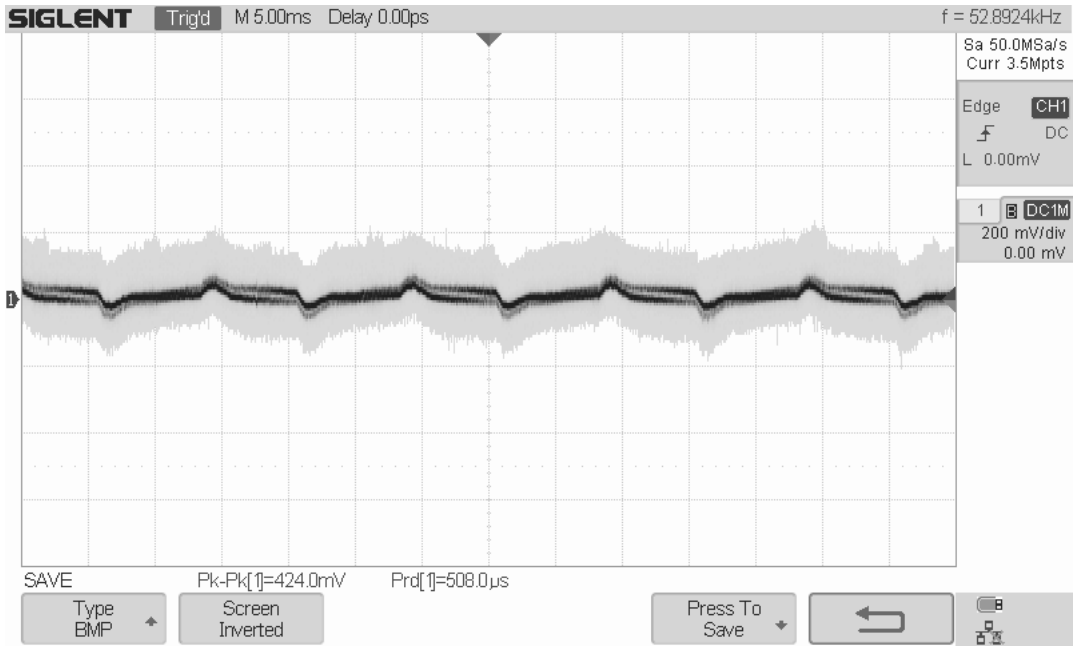


Figure 2.16: Experimental result of the output current for a nonlinear load

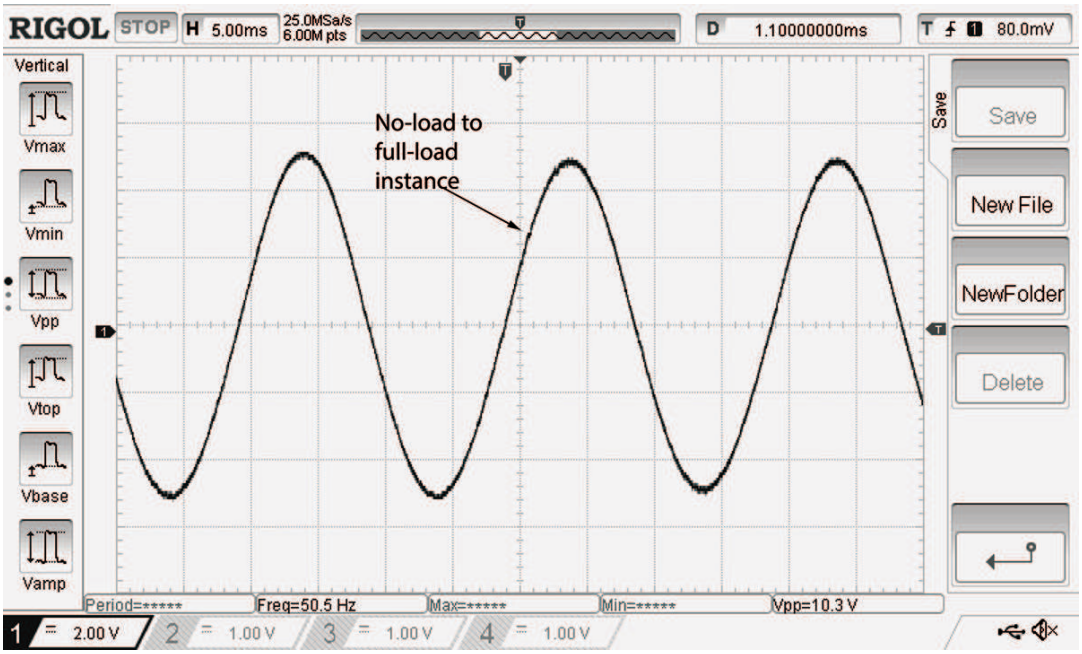


Figure 2.17: Experimental result of the output voltage for no-load to full-load

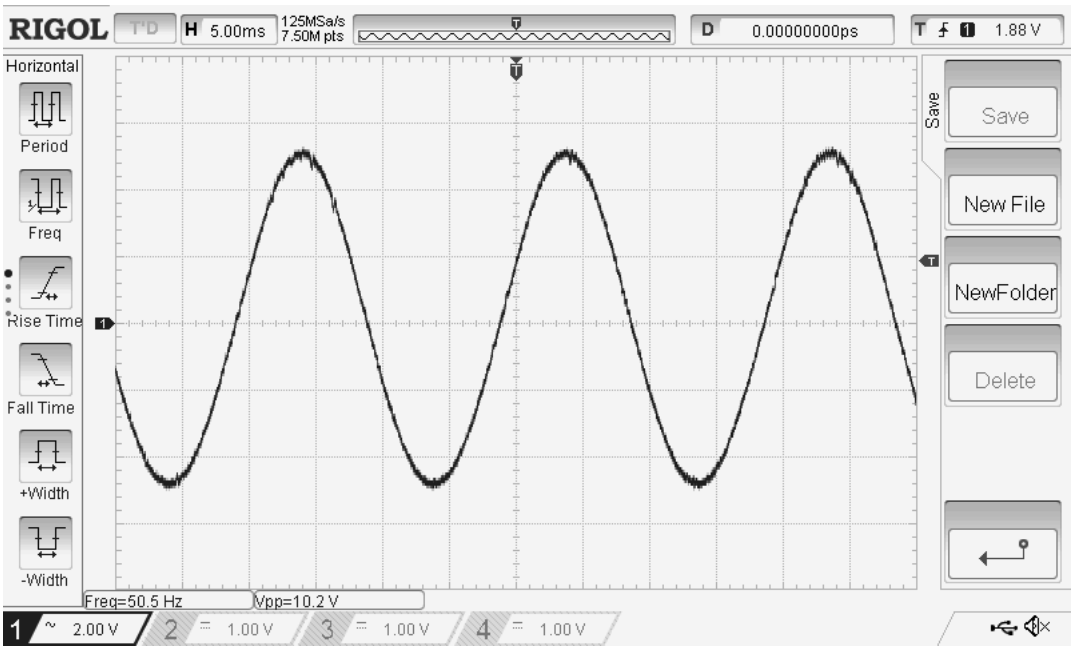


Figure 2.18: Experimental result of the output voltage of ADP controller with parameter mismatch

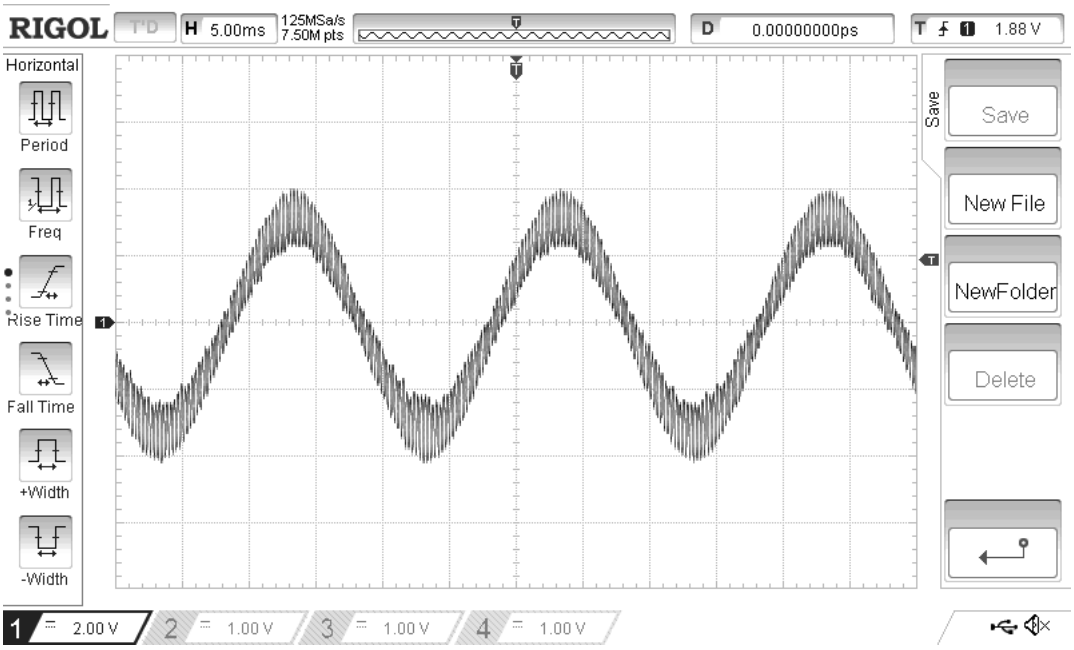


Figure 2.19: Experimental result of the output voltage of PWM controller with parameter mismatch

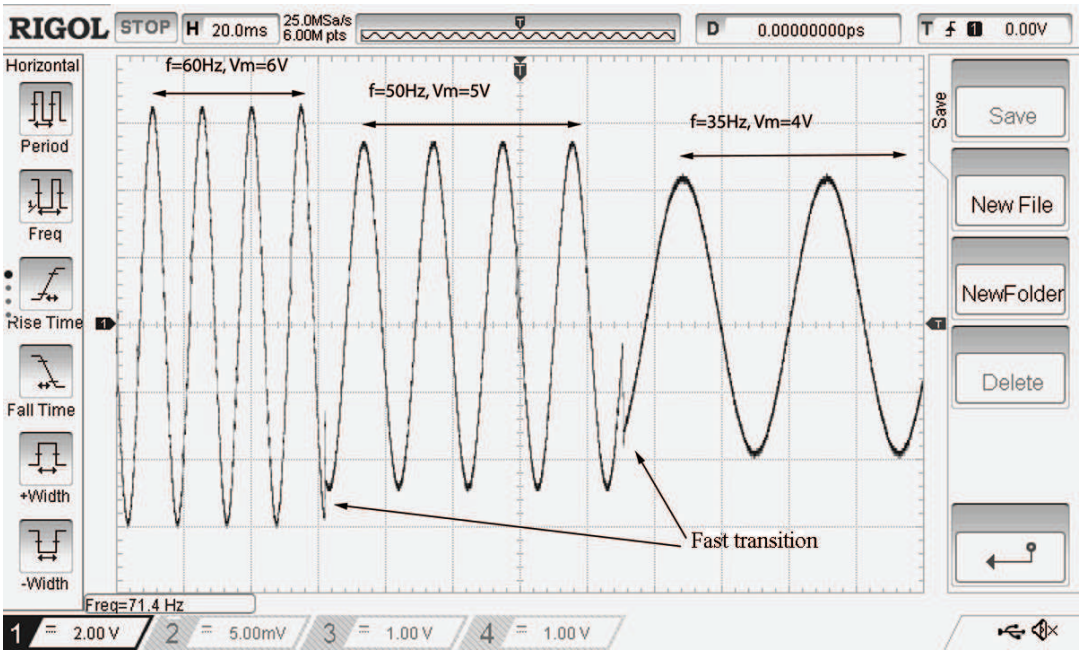


Figure 2.20: Experimental result of the output voltage of ADP controller with variable frequency and amplitude

Chapter 3

Preliminaries on Permanent Magnet Synchronous Motor Control

Due to high performance, reliable operation and simple structure of Permanent magnet synchronous motors (PMSMs), they are widely used in different applications such as robotics, electric vehicles, appliances, etc. In this chapter, after defining Clarke and Park transformations, the PMSM dynamic model is presented. Then space vector pulse width modulation (SVPWM), which is widely used in PMSM control, is introduced. Finally the two common PMSM control approaches, namely field-oriented control (FOC) and direct torque control (DTC) are presented.

3.1. Clark and Park Transform

Let i_a, i_b, and i_c be the instantaneous balanced three-phase currents, [9]. Then,

$$i_a(t) + i_b(t) + i_c(t) = 0. \tag{3.1}$$

current space vector can be represented as

$$\bar{i} = \frac{2}{3}(i_a(t) + i_b(t)e^{j\frac{2\pi}{3}} + i_c(t)e^{j\frac{4\pi}{3}}), \tag{3.2}$$

where the three-phase currents are now the projections of the vector $\bar{i}$ on each winding axes. Other three-phase quantities such as voltages and flux linkages can be shown as a space vector as well.

The space vector defined in (3.2) can be expressed in a Cartesian coordinate system, which is fixed to stator. The real part of this coordinate system (α) is chosen identical to one of the winding axes, such as phase 'a'. The imaginary axis of this coordinate system is shown with β. The real part of the space vector is equal to the direct axis component, i_α, and the imaginary part is equal to the quadrature axis component, i_β. Fig. 3.1 shows the current space vector and its projections on stationary direct and quadrature axes.

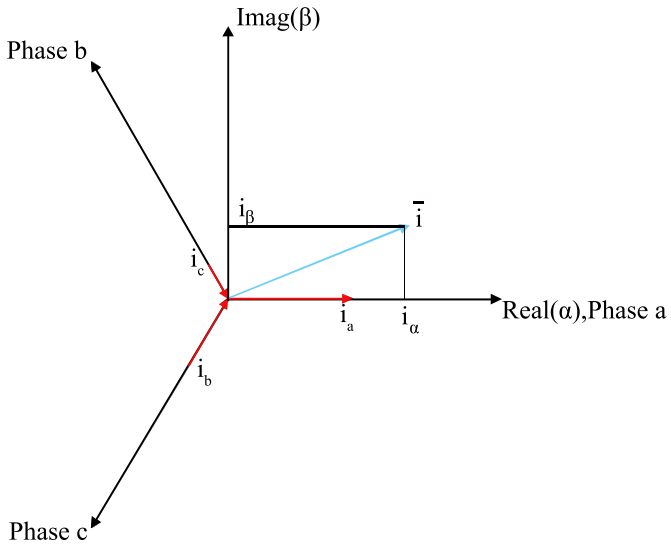


Figure 3.1: Current space vector and its projections

The transformation from three-phase abc to a stationary 2-phase $\alpha\beta$ system is commonly known as Clarke transformation with the following relation

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}, \tag{3.3}$$

Fig. 3.2, shows the transformation of three-phase currents to stationary two-phase currents. Another Cartesian coordinate system is also used to express the three-phase components. This coordinate system is fixed on the rotor and rotates at an angular speed of $\omega_s = \frac{d\theta_s}{dt}$ as shown in Fig. 3.3. θ_s is the angle between the real axis of the stationary reference frame and the real axis of the rotating reference frame (d). The imaginary axis of this coordinate system is shown with q. The

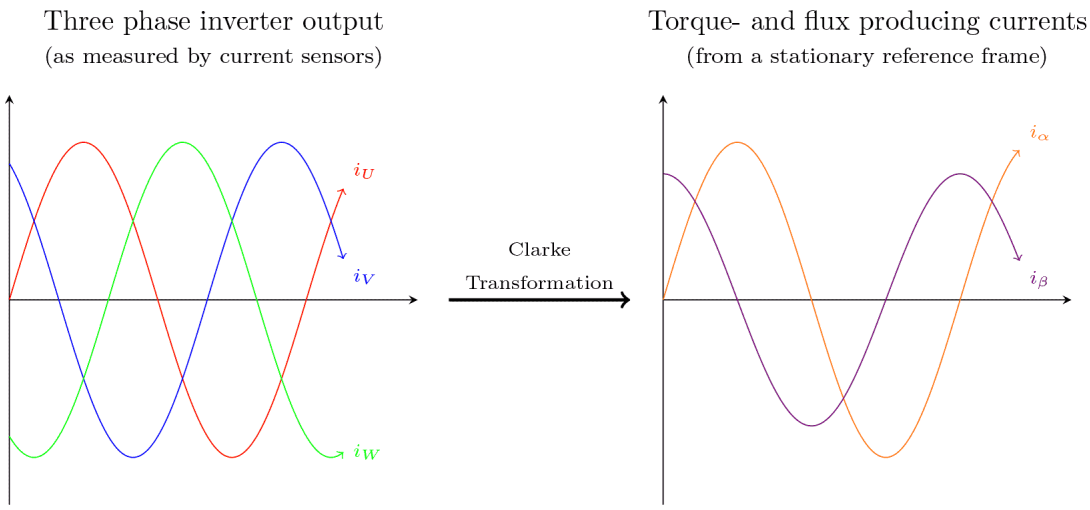


Figure 3.2: transformation of three-phase currents to stationary two-phase currents. (Picture adapted from [73])

transformation from $\alpha\beta$ to dq is called Park transformation with the following relation

$$\begin{bmatrix} i_d \\ i_d \end{bmatrix} = \begin{bmatrix} \cos(\theta_s) & \sin(\theta_s) \\ -\sin(\theta_s) & \cos(\theta_s) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}, \tag{3.4}$$

Fig. 3.4, shows the transformation of stationary two-phase currents to rotating dq current.

3.2. Permanent Magnet Synchronous Motor

A permanent magnet synchronous machine (PMSM) is an electrical machine in which the rotor excitation field is provided by permanent magnets. The stator is constructed in a similar way as in AC induction machines. In order to analyze a PMSM, the most convenient way is to consider the motor parameters on a rotating reference frame fixed on the rotor. This can be done by transforming the three-phase components to dq components using the Clarke and Park transformations in (3.3) and (3.4). Note that in (3.4), θ_s should be the electrical angle of the rotor, which is equal to $P\theta_m$, where P is the number of rotor pole pairs and θ_m is mechanical angle of the rotor. The dynamic model of a PMSM in the dq rotating reference frame is as follows

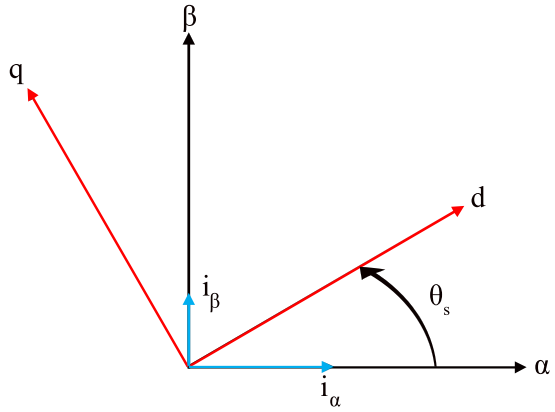


Figure 3.3: dq rotating frame

$$\begin{aligned} v_d &= R_s i_d + L_d \frac{di_d}{dt} - L_q P \omega_m i_q \\ v_q &= R_s i_q + L_q \frac{di_q}{dt} + L_d P \omega_m i_d - \lambda_m P \omega_m, \end{aligned} \tag{3.5}$$

where v_d, v_q, and i_d, i_q are the stator voltage and current in dq reference frame. R_s is the stator winding resistance, L_d, L_q are stator winding inductance in dq coordinates, ω_m is the rotor mechanical speed, and λ_m is the magnetic flux linkage of the rotor permanent magnets. The torque balance equation of the motor is

$$\frac{d}{dt}\omega_m = \frac{1}{J}(\tau_{em} - \tau_f - \tau_L), \tag{3.6}$$

where J is the rotor and load inertia, τ_f, and τ_L are friction and load torques, respectively. The motor electromagnetic torque, denoted with τ_{em}, is given by

$$\tau_{em} = \frac{3}{2} P((L_d - L_q) i_d i_q + \lambda_m i_q), \tag{3.7}$$

Depending on the location of the magnets on the rotor, PMSMs can be divided into two main categories, the surface-mount PMSM (SPMSM) and Interior PMSM (IPMSM). In a typical SPMSM, the magnets are fixed to the exterior of the rotor, however in an IPMSM, the magnets are embedded into the rotor [29]. Fig. 3.5 shows the cross section of these two types of PMSMs.

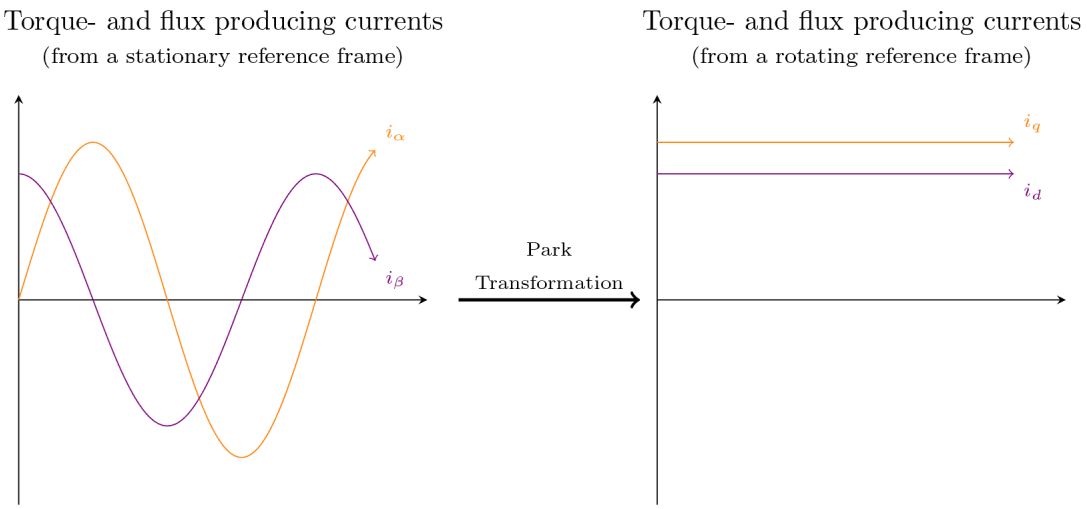


Figure 3.4: transformation of stationary two-phase currents to rotating dq current. (Picture adapted from [73])

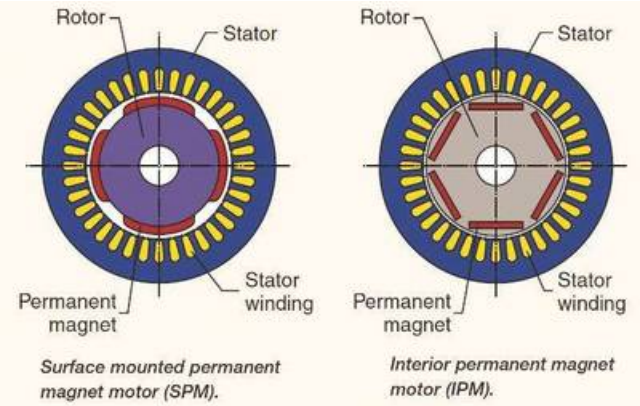


Figure 3.5: surface-mount PMSM and Interior PMSM, (Picture adapted from [29])

The location of magnets in SPMSM and IPMSM leads to different characteristics in these motors. In order to identify the difference between SPMSM and IPMSM, let us define magnetic saliency. Magnetic saliency describes the relationship between the rotor's main flux (d axis) inductance and the main torque-producing (q axis) inductance. The magnetic saliency varies depending on the position of the rotor to the stator field, where maximum saliency occurs at 90 electrical degrees from the main flux axis (d axis). Since the magnetic saliency is limited for SPMSM ($L_d \approx L_q$), it can only generate the *magnetic* torque. However, the magnetic saliency is relatively high in IPMSMs ($L_q > L_d$), therefore they can generate both *magnetic and reluctance* torque, [29].

3.3. Space Vector Pulse Width Modulation

In this section space vector modulation (SVM) is introduced. The contents of this section are from [15]. Let us consider the three-phase inverter in Fig. 1.2. There are six switches in this inverter topology, which can generate eight unique voltage vectors. For instance, if switch S_1 is on, then S_2 is off and the switching state of the first leg is $T_1 = 1$. Conversely, if $T_1 = 0$, then S_1 is off and S_2 is on, [15]. The relation between switching states of the inverter and each voltage phase is

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \frac{V_{dc}}{3} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}, \tag{3.8}$$

The three-phase voltages, voltage vectors and two-phase voltages in stationary $\alpha\beta$ coordinate system corresponding to each switching state are presented in Tab. 3.1

Table 3.1: Voltage vectors generated by a two-level VSI

$\bar{V}_x(T_1,T_2,T_3)$	v_a	v_b	v_c	$\bar{V}_s$	V_α	V_β
$\bar{V}_1(1,0,0)$	$\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}\angle 0°$	1	0
$\bar{V}_2(1,1,0)$	$\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$\frac{2}{3}V_{dc}\angle 60°$	$\frac{1}{3}V_{dc}$	$\frac{\sqrt{3}}{3}V_{dc}$
$\bar{V}_3(0,1,0)$	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}\angle 120°$	$-\frac{1}{3}V_{dc}$	$\frac{\sqrt{3}}{3}V_{dc}$
$\bar{V}_4(0,1,1)$	$-\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}\angle 180°$	-1	0
$\bar{V}_5(0,0,1)$	$-\frac{1}{3}V_{dc}$	$-\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}$	$\frac{2}{3}V_{dc}\angle 240°$	$-\frac{1}{3}V_{dc}$	$-\frac{\sqrt{3}}{3}V_{dc}$
$\bar{V}_6(1,0,1)$	$\frac{1}{3}V_{dc}$	$-\frac{2}{3}V_{dc}$	$\frac{1}{3}V_{dc}$	$\frac{2}{3}V_{dc}\angle 300°$	$\frac{1}{3}V_{dc}$	$-\frac{\sqrt{3}}{3}V_{dc}$
$\bar{V}_7(0,0,0)$	0	0	0	0	0	0
$\bar{V}_8(1,1,1)$	0	0	0	0	0	0

Fig. 3.6 shows the generated voltage vectors corresponding to Tab. 3.1. The area between each voltage vector is called a sector. Suppose a reference voltage vector $\bar{V}_r$ is calculated by the control algorithm. This voltage can be synthesized using the two neighboring active voltage vectors. In order to have a fixed sampling interval, zero vectors are also used. The sampling time T_s is divided

into three time durations, t_x, t_y, and t_0. If the sampling time is T_s, and the reference voltage is in sector 1 as shown in Fig. 3.6, it can be expressed as

$$T_s\bar{V}_r = t_x\bar{V}_1 + t_y\bar{V}_2 + t_0\bar{V}_0, \tag{3.9}$$

This means that, in order to realize a voltage vector located in sector 1, $\bar{V}_1$ is applied for t_x seconds, $\bar{V}_2$ is applied for t_y seconds, and $\bar{V}_0$ is applied for t_0 seconds. The switching time duration for t_x, t_y, and t_0 are calculated as:

$$\begin{aligned} t_x &= \frac{\sqrt{3}|\bar{V}_r|}{V_{dc}} T_s \sin(\frac{\pi}{3} - \alpha) \\ t_y &= \frac{\sqrt{3}|\bar{V}_r|}{V_{dc}} T_s \sin(\alpha) \\ t_0 &= T_s - t_x - t_y \end{aligned} \tag{3.10}$$

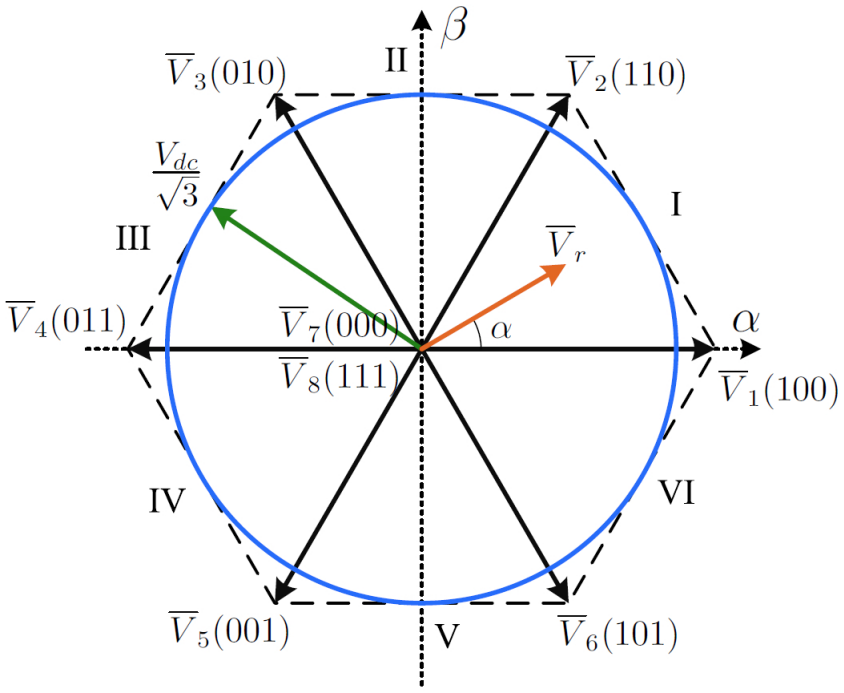


Figure 3.6: Basic voltage vectors and a reference voltage vector in a complex plane, (Picture adapted from [15])

Equation (3.10) can also be used in other sectors. One should only determine in which sector the voltage vector is located and calculate α as the angle between the voltage vector and the immediately previous basic vector. A common switching pattern for two consecutive T_s is presented in Tab. 3.3.

Table 3.3: SVM sequence and timing

Sector	Voltage Vector Sequence	Voltage Vector Timing
I	$\bar{V}_7 \to \bar{V}_1 \to \bar{V}_2 \to \bar{V}_8 \to \bar{V}_2 \to \bar{V}_1 \to \bar{V}_7$	$\frac{t_0}{2} \to t_x \to t_y \to t_0 \to t_y \to t_x \to \frac{t_0}{2}$
II	$\bar{V}_7 \to \bar{V}_3 \to \bar{V}_2 \to \bar{V}_8 \to \bar{V}_2 \to \bar{V}_3 \to \bar{V}_7$	$\frac{t_0}{2} \to t_y \to t_x \to t_0 \to t_x \to t_y \to \frac{t_0}{2}$
III	$\bar{V}_7 \to \bar{V}_3 \to \bar{V}_4 \to \bar{V}_8 \to \bar{V}_4 \to \bar{V}_3 \to \bar{V}_7$	$\frac{t_0}{2} \to t_x \to t_y \to t_0 \to t_y \to t_x \to \frac{t_0}{2}$
IV	$\bar{V}_7 \to \bar{V}_5 \to \bar{V}_4 \to \bar{V}_8 \to \bar{V}_4 \to \bar{V}_5 \to \bar{V}_7$	$\frac{t_0}{2} \to t_y \to t_x \to t_0 \to t_x \to t_y \to \frac{t_0}{2}$
V	$\bar{V}_7 \to \bar{V}_5 \to \bar{V}_6 \to \bar{V}_8 \to \bar{V}_6 \to \bar{V}_5 \to \bar{V}_7$	$\frac{t_0}{2} \to t_x \to t_y \to t_0 \to t_y \to t_x \to \frac{t_0}{2}$
VI	$\bar{V}_7 \to \bar{V}_1 \to \bar{V}_6 \to \bar{V}_8 \to \bar{V}_6 \to \bar{V}_1 \to \bar{V}_7$	$\frac{t_0}{2} \to t_y \to t_x \to t_0 \to t_x \to t_y \to \frac{t_0}{2}$

Fig. 3.7 shows the pulse patterns for three phases, when the voltage vector is located in sector 1. It corresponds to the first row of Tab. 3.3.

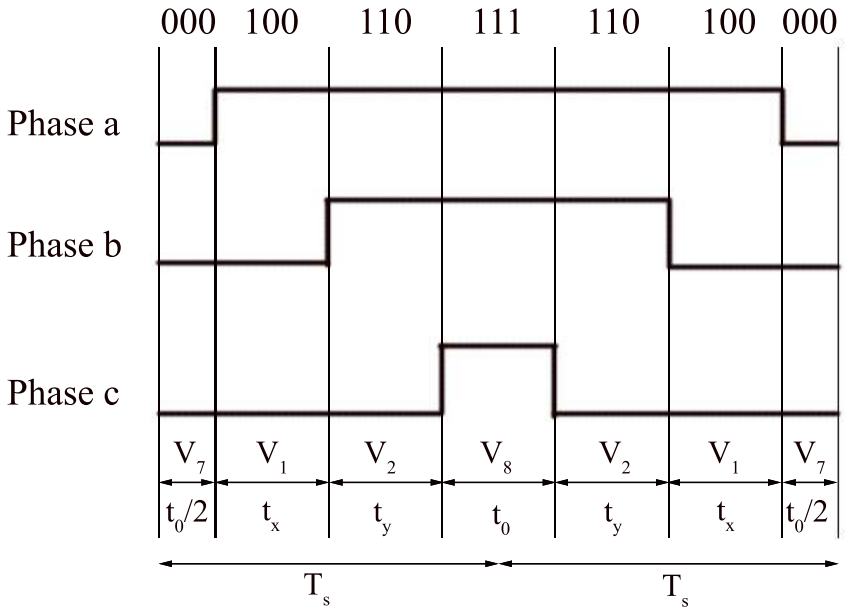


Figure 3.7: Pulse pattern of voltage vector located in sector 1

3.4. Field-Oriented Control

Field-oriented control (FOC) or vector control is one of the most common control methods of PMSMs. In electrical machines, the torque is proportional to the outer product of two magnetic flux linkage vectors, i.e. the rotor and the stator magnetic flux linkage vectors. In a separately excited DC machine, the two magnetic fields are provided by the stator, or field winding and rotor, or armature winding. Also in these machines, a commutation system locates these fields perpendicular to each other, which maximizes the generated torque. In AC machines, the two fields, which generate the torque, are generally neither independent, nor perpendicular. Also there is no mechanical commutator to position these fields, orthogonal with respect to each other. The main objective of FOC is to make the two magnetic fields independent and perpendicular to each other. Therefore, the performance of the AC machine resembles to the performance of the separately excited DC machine. FOC transforms the three-phase currents of the AC machine into two perpendicular currents, which act similar to the field current and armature current in a DC machine, [75].

In this subsection, FOC is presented for surface-mounted PMS motors. Note that in a SPMSM, since ($L_d \approx L_q$), according to (3.7), i_d does not contribute to the torque and the torque is a linear algebraic function of i_q. Therefore, torque control is equivalent to the control of i_q, with a coefficient of $\frac{3}{2}P\lambda_m$. As seen in (3.5), if $i_d = 0$, then i_q is related to v_q with a linear differential equation. Therefore, keeping $i_d = 0$, linearizes the q-axis voltage equation of the machine so a linear controller can be designed to control i_q. Furthermore, if $i_d = 0$, the only current component would become i_q and the maximum torque would be generated in the machine. Fig. 3.8 shows the FOC block diagram. This block diagram is for the speed control of a SPMSM. It is seen that the outer loop is the speed control loop and there are two inner loops. One for controlling i_q and the other one for controlling i_d. Note that as stated before, $i_d^* = 0$.

3.5. Direct Torque Control

This section presents the direct torque control (DTC) as another common approach for PMSM control. The information in this section are gathered from [75]. As mentioned before, in AC machines, the torque can be regarded as the outer product of the stator and rotor flux linkages.

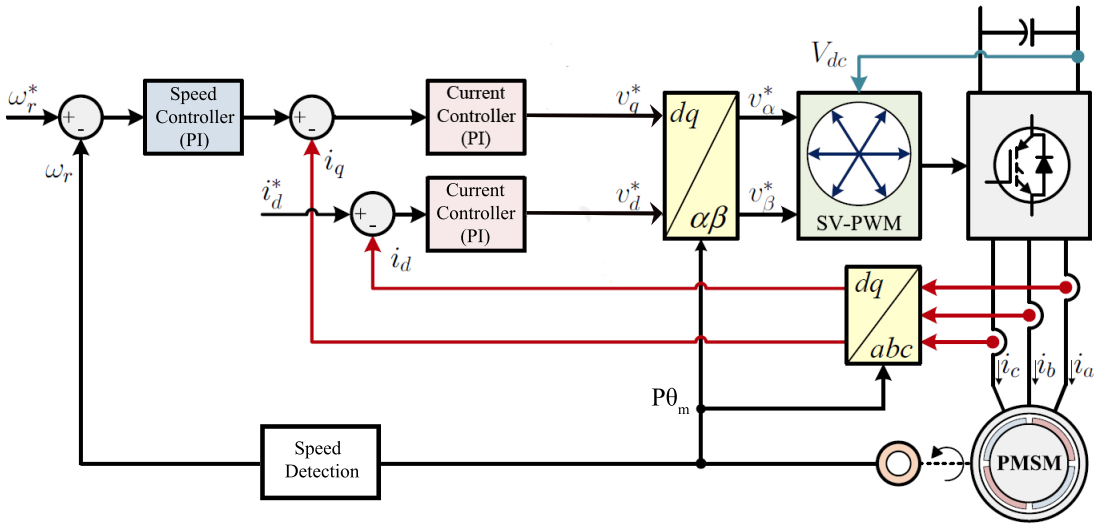


Figure 3.8: Block Diagram of the FOC with SVM, (Picture adapted from [15])

Electromagnetic torque in a SPMSM can be written as

$$\tau_{em} = \frac{3}{2}\frac{P}{L_s}\lambda_m \lambda_s \sin\delta, \tag{3.11}$$

where λ_m is the rotor flux linkage, λ_s is the stator flux linkage, and δ, which is called the load angle, is the angle between these two fluxes.

In a PMSM, the rotor flux linkage depends on magnet poles and is fixed for a specific machine. The magnitude of the stator flux linkage is also assumed to be constant as a constant stator flux is a common practice in many motor control approaches in order to achieve fast dynamics. Therefore, the torque only depends on a single variable δ. Note that λ_m and λ_s are rotating vectors. λ_m rotates with the rotor speed, and since rotor has a large time constant, its speed can not change rapidly. Therefore, the speed of the stator flux should change rapidly in response to the motor command in order to change δ.

The voltage equation of a PMSM in space vector can be written as

$$\bar{V}_s = R_s \bar{i}_s + \frac{d}{dt}\bar{\lambda}_s, \tag{3.12}$$

where $\bar{V}_s$, $\bar{i}_s$, and $\bar{\lambda}_s$ are stator voltage, current, and flux linkage vectors. The first term on the right hand side is the voltage drop on the stator windings and the second term is the induced voltage in stator windings. For high voltage and high speed operation, it is possible to simplify (3.12) and consider stator voltage vector as $\bar{V}_s \approx \frac{d}{dt}\bar{\lambda}_s$, which yields $\Delta\bar{\lambda}_s \approx \bar{V}_s \Delta t$. This simple equation states that a certain flux linkage deviation vector, can be obtained by applying a specific voltage vector to the stator winding for a short time interval. Note that since Δt is a scalar quantity, $\Delta\bar{\lambda}_s$ and $\bar{V}_s$ have the same direction. Fig. 3.9 shows this concept. As mentioned before, the voltage vector $\bar{V}_s$ that causes $\Delta\delta$ is parallel to $\Delta\bar{\lambda}_s$. The voltage vector must be produced by the inverter and applied on the motor for the duration of Δt.

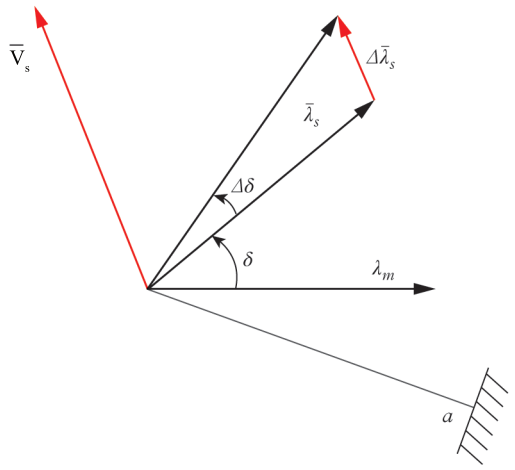


Figure 3.9: Load angle and torque adjustment by voltage vector. (Picture adapted from [75])

It can be proved that the torque dynamics depends on the rotation speed of the stator flux linkage vector with regard to the magnet flux linkage vector, which means changing the load angle δ as fast as possible. Therefore, it is ideal to keep the voltage vector $\bar{V}_s$, perpendicular to $\bar{\lambda}_s$, to get the fastest rotation of $\bar{\lambda}_s$. Note that the three-phase inverter can generate six non-zero voltage vectors as shown in Fig. 3.6. These voltage vectors are not perpendicular to $\bar{\lambda}_s$ in general. Therefore, a compromise is to choose a voltage vector which has an angle to $\bar{\lambda}_s$ as close as possible to 90°. There are always two pairs of voltage vectors that better satisfy this condition. Looking at Fig. 3.10, it is seen that in order to increase δ, to increase the torque, $\bar{V}_3$ or $\bar{V}_4$ are good choices. Also, in order to decrease δ,

to decrease the torque, $\bar{V}_1$ or $\bar{V}_6$ should be selected. Therefore, the desirable voltage is dependent on the sector, in which $\bar{\lambda}_s$ is located.

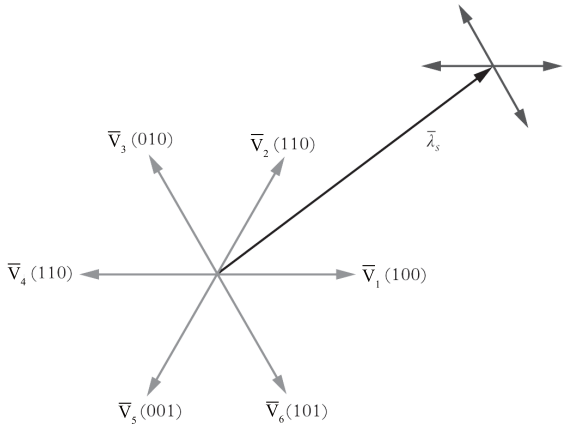


Figure 3.10: Voltage vector options to cause fast dynamics for either increasing or decreasing the developed machine torque. (Picture adapted from [75])

It is also desired to keep the magnitude of $\bar{\lambda}_s$ constant to improve machine dynamics. In Fig. 3.10, if $\bar{V}_3$ and $\bar{V}_4$ are applied to the motor, they increase the torque, but at the same time increase and decrease the magnitude of $\bar{\lambda}_s$, respectively. Similarly, if $\bar{V}_1$ and $\bar{V}_6$ are applied to the motor to decrease the torque, $\bar{V}_1$ increases and $\bar{V}_6$ decreases the magnitude of $\bar{\lambda}_s$. Therefore, it is better to choose a voltage vector that creates the least change in the magnitude of $\bar{\lambda}_s$. In order to limit the changes in the stator flux linkage value, a commanded value of flux linkage a flux linkage band with upper and lower limits are selected. As shown in Fig. 3.11, when a voltage vector is applied to the machine, the flux linkage changes inside the band until it reaches the limits. At this instant, the inverter switches to another voltage vector.

Fig.3.12 shows the block diagram of the DTC method. The torque and flux linkage magnitude commands, T_e^* and λ_s^* are compared with their estimated values and the errors are fed to hysteresis controllers to generate the torque and flux linkage flags, τ and ϕ, respectively. The flags and the stator flux linkage position, are fed to a switching table to determine the suitable voltage vector. An outer loop similar to FOC can be considered for speed control.

Fig.3.13 shows the torque and flux linkage hysteresis controllers. These controllers accept the error as the input and provide the flag of 0 or 1 at the output, depending on the current and previous error values. The hysteresis band of ΔT and $\Delta\lambda$ are selected as the acceptable deviation of the estimated values from the commanded values. If the error is within the band, the controller output will be the same. However, if the input goes beyond the band, the output will change, as follows

$$\begin{cases} \tau = 1, & \text{if } \Delta T_e > \frac{\Delta T}{2} \\ \tau = 0, & \text{if } \Delta T_e < -\frac{\Delta T}{2} \\ \phi = 1, & \text{if } \Delta\lambda_s > \frac{\Delta\lambda}{2} \\ \phi = 0, & \text{if } \Delta\lambda_s < -\frac{\Delta\lambda}{2} \end{cases} \tag{3.13}$$

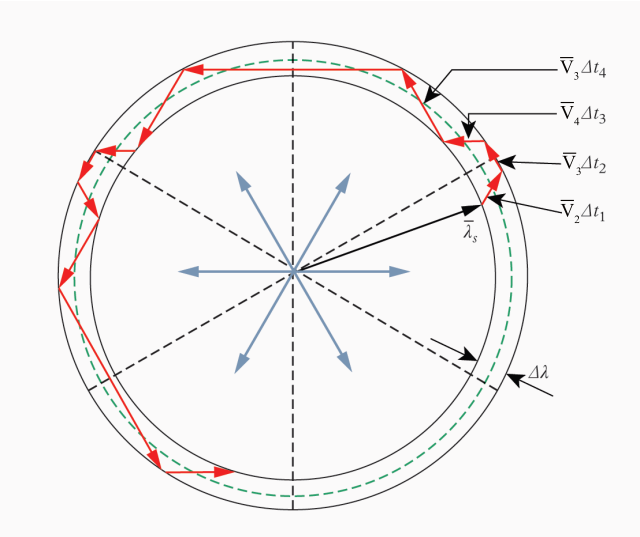


Figure 3.11: Flux linkage band and consecutive voltage vectors in DTC (Picture adapted from [75])

The switching table is presented in Tab.3.5. The inputs to this table are torque and flux flags (τ, ϕ) and the sector number (k).

As seen in Fig.3.12, It is required to estimate the torque and flux of the machine. In order to calculate these parameters, the three-phase currents and voltages are transformed to two-phase fixed coordinates (α, β) and the following relations are used

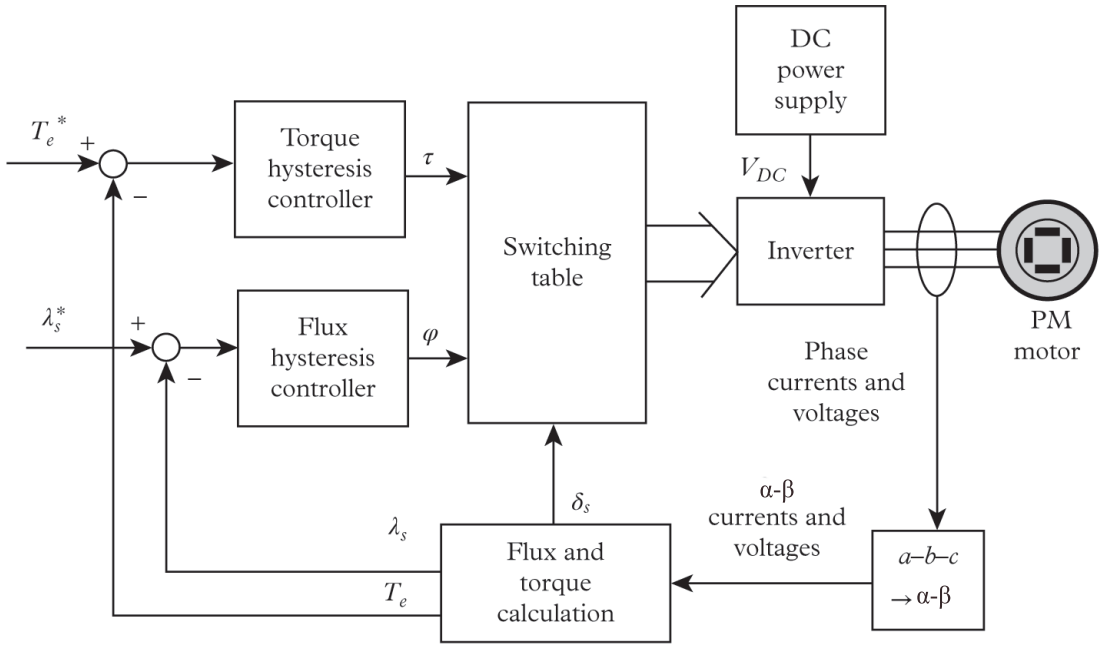


Figure 3.12: DTC block diagram (Picture adapted from [75])

$$\begin{aligned}\lambda_\alpha &= \int_{T_s} V_\alpha - R_s i_\alpha dt \\ \lambda_\beta &= \int_{T_s} V_\beta - R_s i_\beta dt \\ \lambda_s &= \sqrt{\lambda_\alpha^2 + \lambda_\beta^2}\end{aligned} \tag{3.14}$$

$$\delta_s = \tan^{-1}\left(\frac{\lambda_\beta}{\lambda_\alpha}\right) \tag{3.15}$$

$$T_e = \frac{3}{2} P(\lambda_\alpha i_\beta - \lambda_\beta i_\alpha) \tag{3.16}$$

Note that the value of δ_s determines the sector (k), which is used in the switching table in Tab.3.5.

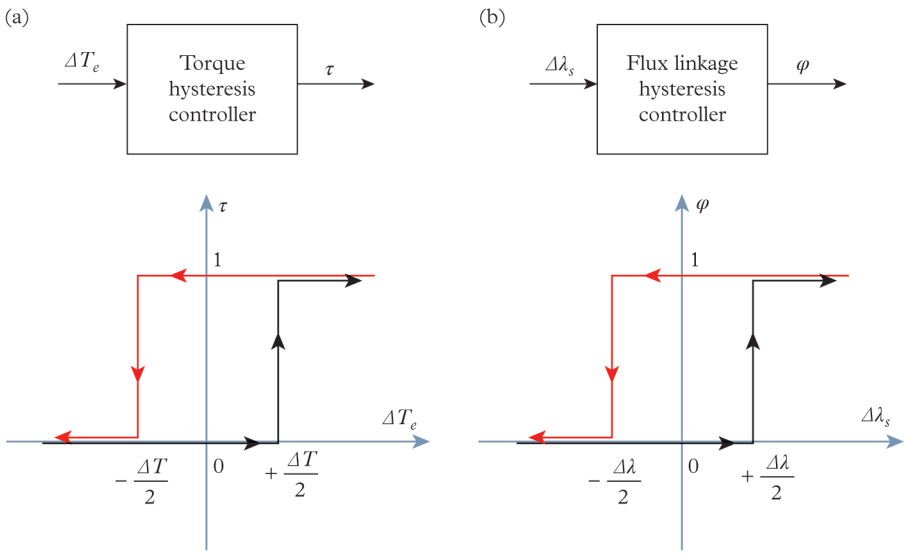


Figure 3.13: Hysteresis controllers: (a) torque controller and (b) flux linkage controller. (Picture adapted from [75])

Table 3.5: DTC Switching Table

τ	ϕ	Voltage Vector
1	1	$\bar{V}_{k+1}$
1	0	$\bar{V}_{k+2}$
0	1	$\bar{V}_{k-1}$
0	0	$\bar{V}_{k-2}$

3.6. Direct Torque Control with Space Vector Modulation

The conventional DTC approach with hysteresis controllers and switching table provides fast machine dynamics and does not need to sense the rotor position (sensorless). However, it suffers from serious drawbacks. Due to the small hysteresis bands, the sampling period should be very short. This sampling period, typically is 10 times shorter than that of FOC. On the other hand, the estimation of torque and flux should be very accurate to get the best result. If wider bands are chosen, to increase the sampling period, then motor torque, flux linkage and current ripples can become high and degrade the machine performance. These issues can be avoided by using PI

controllers instead of hysteresis controllers and using space vector modulation instead of a switching table. The information in this section are gathered from [75]. Fig.3.14 shows the block diagram of DTC-SVM.

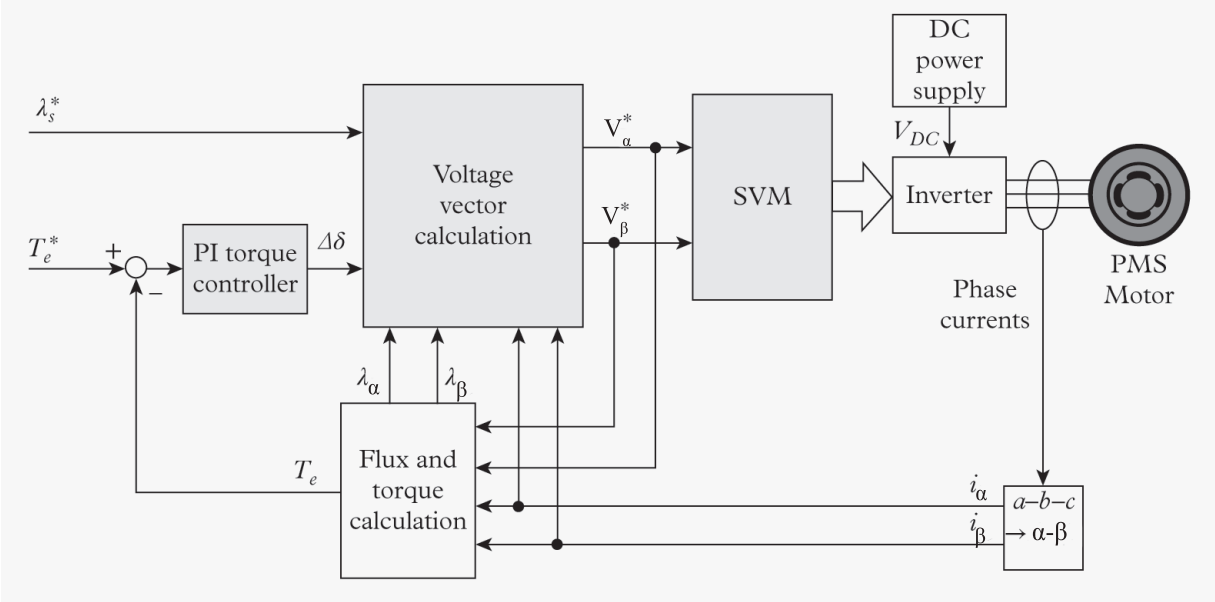


Figure 3.14: DTC-SVM block diagram (Picture adapted from [75])

The reference voltages, V_α^* and V_β^* are calculated as follows

$$\begin{aligned} V_\alpha^* &= R_s i_\alpha + \frac{\Delta\lambda_\alpha}{T_s} \\ V_\beta^* &= R_s i_\beta + \frac{\Delta\lambda_\beta}{T_s} \end{aligned} \tag{3.17}$$

where T_s is the sampling time and

$$\begin{aligned} \Delta\lambda_\alpha &= \lambda_\alpha^* - \lambda_\alpha = \lambda_s^* \cos(\Delta\delta + \delta_s) - \lambda_\alpha \\ \Delta\lambda_\beta &= \lambda_\beta^* - \lambda_\beta = \lambda_s^* \sin(\Delta\delta + \delta_s) - \lambda_\beta \\ \delta_s &= \tan^{-1}\frac{\lambda_\beta}{\lambda_\alpha} \end{aligned} \tag{3.18}$$

Chapter 4

Optimal Torque Control of Permanent Magnet Synchronous Motors Using Adaptive Dynamic Programming

A new approach based on adaptive dynamic programming (ADP) is proposed to control permanent magnet synchronous motors (PMSMs). The objective of this chapter is to control the torque and consequently the speed of a PMSM when an unknown load torque is applied to it. The proposed controller achieves a fast transient response, low ripples and small steady-state error. The control algorithm uses two neural networks, called critic and actor. The former is utilized to evaluate the cost and the latter is used to generate control signals. The training is done once offline and the calculated optimal weights of actor network are used in online control to achieve fast and accurate torque control of PMSMs. This algorithm is compared with field oriented control (FOC) and direct torque control based on space vector modulation (DTC-SVM). Simulations and experimental results show that the proposed algorithm provides desirable results under both accurate and uncertain modeled dynamics.

4.1. Introduction

In recent years, there has been an increasing attention to permanent magnet synchronous motors (PMSMs) because of their undeniable advantages, such as high power density, high torque to inertia ratio and reliability [57, 75]. The performance of a PMSM highly depends on the quality of its control scheme. In many applications such as electric vehicles (EVs) and robotics, it is necessary to achieve precise control of the motor. The control should be satisfactory in the presence of system parameter uncertainties as well as external disturbances.

Field oriented control (FOC) and direct torque control (DTC) are the two main control approaches of alternate current (AC) servo drives. A typical FOC scheme consists of two inner current loops and one outer speed loop. Proportional-integral (PI) controllers are commonly used to regulate

the motor currents. In order to avoid large overshoots, the bandwidth of these current controllers is limited, which leads to the slow dynamic response of the motor, [54]. Furthermore, the PI gains play a crucial role in the steady state behavior of the motor under FOC, therefore fine tuning of the gains is necessary [5, 51]. In order to control a PMSM through power inverters, typically space vector modulation (SVM) is used to realize the appropriate voltage, which is applied to the motor. This method can provide voltage vectors with adjustable amplitude and phase, [75].

DTC, on the other hand, utilizes another approach for the control. In classical DTC, based on two hysteresis comparators and a switching table, a suitable voltage vector is applied on the motor for the whole sampling time [91]. Although DTC provides a faster dynamic response compared to FOC, it has major disadvantages such as increased torque and stator flux ripples, [54]. Furthermore, DTC requires high sampling frequency for digital implementation of the controller, which in turn, demands more powerful digital signal processors (DSPs) and increases the cost of implementation [62]. In recent years, there has been many studies to address the disadvantages of the classical DTC, [77]. One of these approaches utilizes DTC with SVM. Unlike DTC, which uses one of the available fixed voltage vectors, with a fixed amplitude and phase, for the duration of the control cycle, any arbitrary voltage vector can be generated and applied to the motor, when DTC is augmented with SVM (i.e., DTC-SVM). This voltage vector, which has an adjustable amplitude and phase, is generated using multiple vectors during the sampling time. This approach leads to a reduction in torque and flux ripples [33, 44, 45].

Recently, there has been research on more advanced control approaches for PMSMs, such as sliding mode control (SMC) [12, 53, 84], model predictive control (MPC) [6, 10, 54, 70, 83], adaptive control [30, 61], fuzzy control [8, 17], and neural network control [14, 86]. Each of these controllers offers some improvement in the performance of PMSMs. In [12], a digital sliding mode controller is designed to track the desired motor speed, and a digital observer is utilized to estimate the rotor position and velocity. The authors in [54] have designed a model predictive controller for direct speed control of a PMSM. Although the results show great improvement, one disadvantage of this approach is the need for load torque observer. It may be noted that the load on the motor is typically unknown and varying, in reality. Therefore, it is desired to develop a controller with good

robustness toward an uncertain load. In [10], the authors have pursued an MPC-based approach in which, instead of applying one voltage vector during the control cycle, they apply two voltage vectors and calculate their respective duty cycle, i.e. the percentage of sampling period for which, a specific voltage vector is applied. Since in this approach, the torque and flux need to be estimated, this approach is not relatively robust to parameter uncertainties. An adaptive speed controller is proposed in [61] in which, the controller is divided into two parts, one for stabilization of the error dynamics and the other one for dealing with parameter uncertainties. An artificial neural network (ANN) is used in [86] to control the speed of a PMSM. In this approach, the inner current loops remain like classical FOC, however instead of a PI controller for the outer loop speed controller, an ANN is used to generate the reference current.

Recently, adaptive dynamic programming (ADP) has been used extensively in different applications to solve different optimal control problems [36, 39, 72, 92]. Usually, there are two neural networks in an ADP design. The first one is called the 'critic', which approximates the value function (i.e. optimal cost-to-go) and the second neural network is called the 'actor', which generates the optimal control based on the optimal value function and system states. ADP also has been used in the control of PMSMs in [50, 81]. In [81], a single artificial neural network based on ADP is designed which substitutes the outer loop PI speed controller in classical FOC. There are some disadvantages to this approach. First, the inner current loops are still FOC-base PI controllers, therefore the dynamic response of the control is not fast compared to other approaches such as DTC. Secondly, at each sampling time, there is a need for three consecutive speed error values, instead of only the error at that specific sampling time. In a simultaneous but independent research, the authors in [50], have designed a neural network controller which substitutes the inner loop PI current controllers in the classical FOC to deal with a decoupling inaccuracy issue of FOC. This controller has some relative robustness to parameter uncertainties.

The proposed approach in this research, for PMSM control, consists of two steps. The first step is done offline, such that based on a cost function and the system dynamics, the critic and actor network weights are calculated using value iteration (VI), [22]. Then the actor weights are used in the online control to find the suitable control input in a feedback form. Therefore, the computational

load in the online control stage is as low as evaluating a few algebraic functions (a feed forward of the actor). The control inputs are the voltages applied to the SVM block.

The contribution of this work utilizing ADP for optimal control of PMSMs. This controller leads to a fast dynamic response as well as desirable steady state performance. The strength of this controller is observed more clearly when there are parameter uncertainties and load disturbances in the system. Moreover, after doing the offline training, the calculation load of this controller is as low as evaluating a polynomial function, which is extremely low compared to many other methods, including DTC or MPC. The low complexity of online implementation of this controller makes it a potential candidate for many applications. The proposed controller is compared with classical FOC and DTC-SVM, both in simulation and experiment, to show its superior performance.

As for organization of this chapter, Section 4.2, provides the dynamic equations of a PMSM. Afterwards, the optimal control using ADP in its general form is formulated in Section 4.3. In Section 4.4, the proposed ADP-based controller is simulated on a PMSM. The proposed controller is implemented on a physical prototype and the experimental results are provided in Section 4.5, followed by concluding remarks in Section 4.6.

4.2. Dynamics of PMSM

It is common to write the machine model of a PMSM in the *dq* synchronous coordinates [75]. The parameters in *dq* reference frame can be obtained from *abc* parameters using Park-Clark transform as follows

$$\begin{bmatrix} f_d \\ f_q \\ f_0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta_e) & \cos(\theta_e - \frac{2\pi}{3}) & \cos(\theta_e + \frac{2\pi}{3}) \\ -\sin(\theta_e) & -\sin(\theta_e - \frac{2\pi}{3}) & -\sin(\theta_e + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_a \\ f_b \\ f_c \end{bmatrix}, \tag{4.1}$$

where f can be a phase voltage, a phase current or a phase flux linkage. Also θ_e is the rotor *electrical angle* between phase "a" of the stationary *abc* reference frame and the "d-axis" of the

rotor reference frame and is calculated as

$$\theta_e = P\theta_m, \tag{4.2}$$

where θ_m is the rotor *mechanical angle* and P is the number of pole pairs. After transforming motor voltages and currents from *abc* to *dq* coordinates, the motor dynamic equations are obtained as

$$\dot{\bar{x}} = \frac{d}{dt}\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} L_d^{-1}(-R_s i_d + L_q P \omega_m i_q) \\ L_q^{-1}(-R_s i_q - L_d P \omega_m i_d - \lambda_m P \omega_m) \end{bmatrix} + \begin{bmatrix} L_d^{-1} & 0 \\ 0 & L_q^{-1} \end{bmatrix} \begin{bmatrix} v_d \\ v_q \end{bmatrix} = \bar{f}(\bar{x}) + \bar{g}\bar{u} \tag{4.3}$$

where v_d, v_q, and i_d, i_q are the stator voltage and current in *dq* reference frame. R_s is the stator winding resistance, L_d, L_q are stator winding inductance in *dq* coordinates, ω_m is the rotor mechanical speed, and λ_m is the magnetic flux linkage of the rotor permanent magnets. The torque balance equation of the motor is

$$\frac{d}{dt}\omega_m = \frac{1}{J}(\tau_{em} - \tau_f - \tau_L), \tag{4.4}$$

where J is the rotor and load inertia, τ_f, and τ_L are friction and load torques, respectively. The motor electromagnetic torque, denoted with τ_{em}, is given by

$$\tau_{em} = \frac{3}{2}P((L_d - L_q)i_d i_q + \lambda_m i_q), \tag{4.5}$$

Finally, it may be mentioned that $\bar{x}$, denoting the state vector in (4.3), is composed of i_d and i_q. Moreover, the control vector is denoted with $\bar{u}$ and is composed of voltages v_d and v_q.

4.3. Optimal Control Using ADP

In this section, the optimal control problem is formulated and its solution using ADP is presented. Let the infinite-horizon cost function, subject to minimization, be given by

$$J = \sum_{k=0}^{\infty} \gamma^k (Q(x_k) + u_k^T R u_k), \tag{4.6}$$

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the system states with dimension n, and the control vector with dimension m, respectively. Moreover, $Q : \mathbb{R}^n \to \mathbb{R}_+$ penalizes the states and $R \in \mathbb{R}^{m\times m}$ penalizes the control. Furthermore, $\gamma \in (0,1]$ is the discount factor, which is used to ensure the boundedness of the cost. The discrete-time nonlinear dynamics is defined as

$$x_{k+1} = f(x_k) + g(x_k)u_k, k \in \{0,1,2,...\}, x(0) = x_0, \tag{4.7}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function which represents the internal dynamics of the system and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$, is the input gain function. The objective is to find the sequence of 'optimal' control, denoted with $u_k^*, k \in \{0,1,2,...\}$ such that the cost function in (4.6) is minimized subject to system dynamics in (4.7).

One can write the cost-to-go from current time to infinity, as a function of current state and future decisions, denoted with $V(.,.)$, as

$$V(x_k, \{u_h\}_{h=k}^{\infty}) = \sum_{h=k}^{\infty} \gamma^k (Q(x_h) + u_h^T R u_h). \tag{4.8}$$

It is possible to write the above equation in a recursive form as

$$\begin{aligned} V(x_k, \{u_h\}_{h=k}^{\infty}) &= Q(x_k) + u_k^T R u_k + \gamma V(x_{k+1}, \{u_h\}_{h=k+1}^{\infty}) \\ &= Q(x_k) + u_k^T R u_k + \gamma V(f(x_k) + g(x_k)u_k, \{u_h\}_{h=k+1}^{\infty}), \end{aligned} \tag{4.9}$$

Function $V^*(x_k)$ is called the *value function* which is the optimal cost-to-go from current state at current time to infinity, if the optimal control sequence is applied on the system. Considering the relation given by (4.9), one can find the value function and optimal control sequence based on Bellman principle of optimality [38] as follows

$$\begin{aligned} V^*(x_k) &= inf_{\{u_h\}_{h=k}^{\infty}} (V(x_k, \{u_h\}_{h=k}^{\infty})) \\ &= inf_{u_k \in \mathbb{R}^m} \Big(Q(x_k) + u_k^T R u_k + \gamma V^*(f(x_k) + g(x_k)u_k) \Big), \forall x_k \in \mathbb{R}^n, \end{aligned} \tag{4.10}$$

$$u^*(x_k) = arginf_{u_k \in \mathbb{R}^m} \Big(Q(x_k) + u_k^T R u_k + \gamma V^*(f(x_k) + g(x_k)u_k) \Big), \forall x_k \in \mathbb{R}^n, \tag{4.11}$$

which leads to

$$u^*(x) = -\frac{1}{2}\gamma R^{-1} g(x)^T \nabla V^*(f(x) + g(x)u^*(x)), \forall x \in \mathbb{R}^n, \tag{4.12}$$

$$V^*(x) = Q(x) + u^*(x)^T R u^*(x) + \gamma V^*(f(x) + g(x)u^*(x)), \forall x \in \mathbb{R}^n, \tag{4.13}$$

where $\nabla V(x)$ is defined as $\partial V(x)/\partial x$. Therefore, theoretically the Bellman equation gives the solution to the optimal control problem. However, because of the curse of dimensionality [38], it is not practically possible to solve this optimal control problem for most nonlinear systems. One approach of solving this optimal control problem is by the value iteration (VI)-based ADP. In this approach, the value function and optimal control are approximated as functions of system states in a compact set, which is called *region of interest*, Ω. The *value update* and *policy equation* relations are given as

$$V^{i+1}(x) = Q(x) + u^i(x)^T R u^i(x) + \gamma V^i(f(x) + g(x)u^i(x)), \forall x \in \Omega, \tag{4.14}$$

$$u^i(x) = -\frac{1}{2}\gamma R^{-1} g(x)^T \nabla V^i(f(x) + g(x)u^i(x)), \forall x \in \Omega, \tag{4.15}$$

The approximation is found by starting from an initial guess for the value function as $V^0(x), \forall x \in \Omega$. Then one uses (4.15) to find $u^0(x), \forall x \in \Omega$. At the next step, (4.14) is utilized to find $V^1(x), \forall x \in \Omega$ and so on. The process is repeated until the iterations converge. There are two points regarding this approach. First, it is important to analyze the convergence condition. In other words, what should be the initial guess of $V^0(x), \forall x \in \Omega$, such that the iterations converge to an optimal value? This topic is investigated in [22] and it is shown that if $V^0(.)$ is smooth and $0 \leq V^0(x) \leq Q(x), \forall x \in \Omega$, then the iterations converge to the optimal solution. The second point regarding the iterations, is solving (4.15). It is seen that on two sides of this equation, $u^i(x)$ appears. Therefore, a system of

nonlinear equations with 'm' equations and 'm' unknowns should be solved. It is proposed in [22] that in order to solve this system of nonlinear equations, one can use another set of iterations with index 'j' as follows

$$u^{i,j+1}(x) = -\frac{1}{2}\gamma R^{-1} g(x)^T \nabla V^i(f(x) + g(x)u^{i,j}(x)), \forall x \in \Omega, \tag{4.16}$$

Therefore, in order to find $u^i(x)$, one can start with a random initial guess for $u^{i,0}(x)$ and then iterate through (4.16) until convergence. It is proved in [22] that under the following conditions, this iteration will converge:

) The norm of matrix R^{-1} is small enough.

) The norm of matrix valued function $g(x)$ is small enough, $\forall x \in \Omega$.

Note that continuous-time state equations are utilized to realize a dynamical system. The presented approach is based on discrete-time dynamics, therefore the system equations are discretized with a sampling time. If the sampling time is small enough, then the two conditions for the convergence of $u^i(x)$ can be satisfied.

In order to implement this approach, the critic and actor neural networks are utilized. For these two networks, linear-in-weight neural networks are used as follows

$$V(x) \simeq W_c^T \phi(x), \forall x \in \Omega, \tag{4.17}$$

$$u(x) \simeq W_a^T \sigma(x), \forall x \in \Omega, \tag{4.18}$$

where $\phi : \mathbb{R}^n \to \mathbb{R}^{n_c}$ and $\sigma : \mathbb{R}^n \to \mathbb{R}^{n_a}$ are the basis functions and positive integers n_c and n_a are the number of neurons in the critic and actor, respectively. Finally, $W_c \in \mathbb{R}^{n_c}$ and $W_a \in \mathbb{R}^{n_a \times m}$ denote the network weights, which will be determined through the training. Note that for each iteration of value function $V^i(x)$, a corresponding critic weight W_c^i is calculated.

Therefore, in order to solve an optimal control problem with ADP and using critic and actor networks, one starts with choosing a large number of random states from the region of interest. These random states will be used for training the critic and actor networks. The training algorithm

starts with initializing the value function $V^0(x)$. This initialization should be done for all the randomly selected states. The critic weights, W_c^0, can be obtained by using least square method applied on the input-target pairs $(x, V^0(x))$. At the next step, optimal control corresponding to $V^0(x)$ should be calculated. This is done by initializing $u^{0,0}(x)$ and iterating (4.16) until $u^0(x)$ is achieved. After calculating $u^0(x)$ for all the randomly selected states, (4.14) is used to calculate $V^1(x)$, which leads to the calculation of W_c^1. If the conditions of convergence, which are stated above, are met, then the iterations will converge after κ iterations. The optimal critic weight W_c^* is calculated from the input-target pair $(x, V^\kappa(x))$. The final step is to calculate the optimal actor weights, W_a^*, which is obtained from the input-target pair $(x, u^\kappa(x))$. In the online control stage, W_a^* is used to find the optimal control $u^*(x)$ at each sampling time. Interested readers are referred to [22, 23, 87] for more details on stability and convergence analysis.

4.4. Implementation on PMSM

In this section, the presented ADP-based optimal controller is applied on a Surface Mount-PMSM and its performance is compared with FOC and (PI-based) DTC-SVM, [44]. Also relative robustness of ADP-based controller under parameter uncertainties is shown by some simulations and it is compared with robustness of FOC and DTC-SVM. Fig. 4.1 shows the control block diagram and the PMSM parameters are given in table 4.1.

4.4.1. Neural Network Training

The first step in training the neural networks is defining the cost function. As seen in (4.6), the cost function has two terms, $Q(x)$ and R. Our objective is tracking torque, while guaranteeing maximum torque per ampere (MTPA) criteria. Motivated by [65], $Q(x)$ can be considered as

$$Q(x_k) = K_1(\tau_{em}(k) - \tau_{em}^*(k))^2 + K_2 i_d(k)^2, \tag{4.19}$$

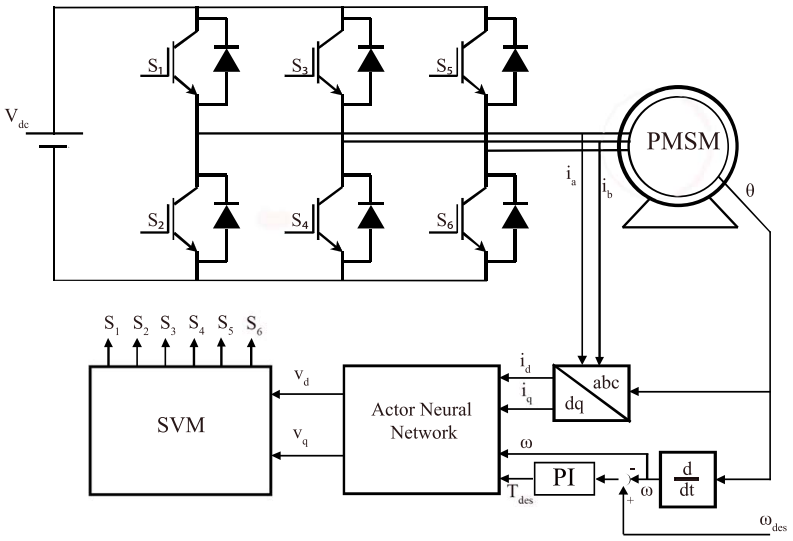


Figure 4.1: PMSM control block diagram using actor neural network

the constant matrix R is utilized to penalize the control inputs, which are v_d and v_q.

$$R = K_3 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{4.20}$$

where K_1, K_2, and $K_3 \in \mathbb{R}$ are design parameters. Design parameter K_1 is the weight penalizing the error between the reference and actual torque, K_2 penalizes i_d, serving the purpose of maximizing MTPA, and K_3 penalizes the control input. There are two important points to be considered in training. First, as seen in (4.19), the objective is tracking the reference torque, therefore this value is needed to find the control signals. In order to consider the effect of reference torque on the control signal, the inputs to the critic and actor are augmented, by adding the reference torque to system states. This way the controller can "learn" how to react when the reference torque changes. Secondly, according to equations (4.14) and (4.15), the right hand side needs the next state x_{k+1}. It is observed in (4.3) that in order to find the next state, the speed ω_m is needed. Therefore, ω_m is added to the inputs of critic and actor as an "exogenous" input. Therefore, the optimal value function and optimal control are functions of discrete-time states ($x = [i_d, i_q]^T$), reference torque

Table 4.1: Motor and Control System Parameters

Parameter	Value
Motor type	Surface Mount-PMSM
Pole pairs, P	5
Permanent magnet flux, λ_m	0.015 Wb
Stator resistance, R_s	1.2 Ω
Stator inductance, $L_d = L_q = L_s$	0.003 H
Rated speed, n_{rated}	3000 rpm
Rated torque, τ_{rated}	0.64 $N.m$
Rated output, P_{out}	0.2 kW
Rated current, I_{rated}	2.5 $A(rms)$
Maximum speed, n_{max}	6000 rpm
Maximum torque, τ_{max}	1.91 $N.m$
Maximum current, I_{max}	7 $A(rms)$
DC bus voltage, V_{dc}	100 V
Rotor moment of inertia, J	30×10^{-6} $kg.m^2$
Sampling time, T_s	40 μS

(τ^*_{em}), and motor mechanical speed (ω_m) . Equations (4.14), (4.15), (4.17), (4.16), and (4.18) are modified as

$$\begin{aligned} V^{i+1}(x,\tau^*_{em},\omega_m) &= Q(x,\tau^*_{em}) + u^i(x,\tau^*_{em},\omega_m)^T R u^i(x,\tau^*_{em},\omega_m) \\ &\quad + \gamma V^i(f(x,\omega_m) + g u^i(x,\tau^*_{em},\omega_m)), \\ &\quad \forall [x^T,\tau^*_{em},\omega_m]^T \in \Omega, \end{aligned} \tag{4.21}$$

$$u^i(x,\tau^*_{em},\omega_m) = -\frac{1}{2}\gamma R^{-1} g^T \nabla V^i(f(x,\omega_m) + g u^i(x,\tau^*_{em},\omega_m)), \forall [x^T,\tau^*_{em},\omega_m]^T \in \Omega, \tag{4.22}$$

$$u^{i,j+1}(x,\tau^*_{em},\omega_m) = -\frac{1}{2}\gamma R^{-1} g(x)^T \nabla V^i(f(x,\omega_m) + g u^{i,j}(x,\tau^*_{em},\omega_m)), \forall [x^T,\tau^*_{em},\omega_m]^T \in \Omega, \tag{4.23}$$

$$V(x,\tau^*_{em},\omega_m) \simeq W_c^T \phi(x,\tau^*_{em},\omega_m), \forall [x^T,\tau^*_{em},\omega_m]^T \in \Omega, \tag{4.24}$$

$$u(x,\tau_{em}^*,\omega_m) \simeq W_a^T \sigma(x,\tau_{em}^*,\omega_m), \forall [x^T,\tau_{em}^*,\omega_m]^T \in \Omega, \tag{4.25}$$

As mentioned in Section 4.3, the region of interest (Ω) for the states should be selected for the training. If elements of the input vector to the networks assume values within comparable ranges, the approximations will have better results, [42]. Therefore, the maximum values of current, speed and torque are used to normalize the states

$$i_d = I_{max}\tilde{i}_d, i_q = I_{max}\tilde{i}_q, \tau_{em}^* = \tau_{max}\tilde{\tau}_{em}^*, \omega_m = \omega_{m_{max}}\tilde{\omega}_m, \tag{4.26}$$

where the normalized quantities are denoted by ' $\sim$ ' notation. Therefore, the region of interest is selected as

$$\Omega = \{[\tilde{i}_d,\tilde{i}_q,\tilde{\omega}_m,\tilde{\tau}_{em}^*]^T \in \mathbb{R}^: -1.5 \leq \tilde{i}_d,\tilde{i}_q,\tilde{\omega}_m,\tilde{\tau}_{em}^* \leq 1.5\}. \tag{4.27}$$

This selection makes sure that the maximum value of the normalized states, which is 1, is well inside the training set, therefore better generalization is achieved near the boundaries of the variation of the states.

Since linear-in-weight neural networks are used for critic and actor, basis functions ϕ and σ should be chosen in (4.17) and (4.18). Motivated by Weierstras approximation theorem, the basis functions are chosen as

$$\begin{aligned}
&\phi(\eta) = [1,\eta^T,(\eta\otimes\eta)^T,(\eta\otimes(\eta\otimes\eta))^T]^T,\\
&\sigma(\eta) = [1,\eta^T,(\eta\otimes\eta)^T]^T,\\
&\eta = [x,\tau_{em}^*,\omega_m],
\end{aligned} \tag{4.28}$$

where $\eta\otimes\eta$, is the non-repeating polynomials built from multiplying elements of vector η by those of η. Since $\eta \in \mathbb{R}^4$, there is 35 neurons for the critic network and 15 neurons for the actor network. These basis function are design parameters, therefore, one can make another choice. The number of 10000 random states are selected from the region of interest to do the batch learning algorithm [22]. The training is done by $K_1 = 30$, $K_2 = 0.5$, $K_3 = 100$, and $\gamma = 0.5$. Fig. 4.2 shows the weights converging after 12 iterations, which took almost 45 seconds on a desktop computer with Intel Core

i-5-6500, 3.2 GHz processor, and 16 GB of RAM, running MATLAB 2018a.

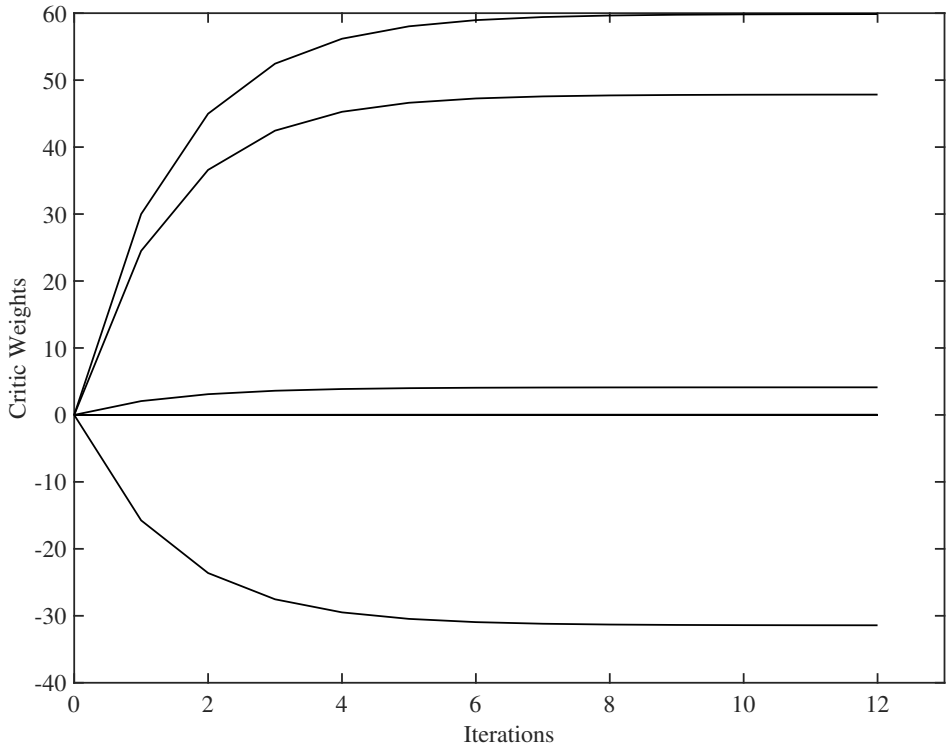


Figure 4.2: History of critic weights during learning iterations

After training and obtaining the weights, the actor weights are calculated using the least square method. The history of critic weights and optimal actor weights can be found in [2]. Thanks to the obtained actor weights, the only computations required in online stage of the control is to calculate $W_a^* \sigma(\eta)$, which will lead to control signals v_d and v_q. These voltages are then applied on the motor by the SVM block. The pseudocodes for offline actor weight calculation and online control are brought in Appendix A.

4.4.2. Comparative simulation

In this section, the calculated actor weights in previous section are used to do simulations on the motor and compare the results with FOC and PI-based DTC-SVM. The PI gains of the speed loop are selected to be identical for all three control algorithms, so that the difference in the performance

is only due to the difference in their torque tracking capability. In the first simulation, the motor shaft rotates at its nominal speed, i.e., 3000 rpm, and a load torque is applied on it. Fig. 4.3 shows the motor speed with three control algorithms, when a load torque of 0.6 $N.m$ is applied on the motor at $t = 1s$. It is seen that the results for three controllers are very close but ADP is showing a slightly better performance, both in reaching the desired speed from standstill and after the load torque is applied. These two parts of the figure are magnified so that the transient response of each algorithm can be seen clearly.

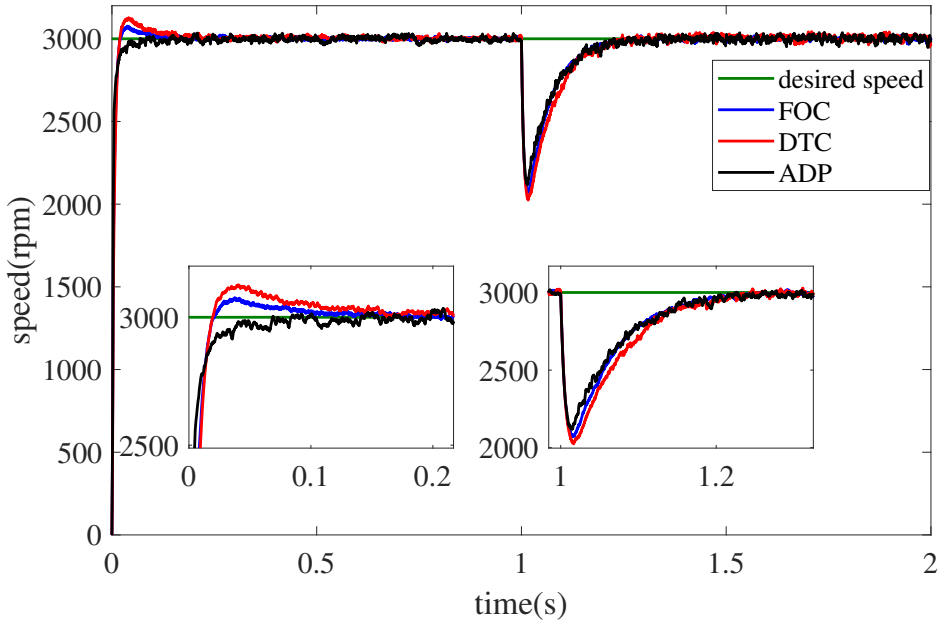


Figure 4.3: Speed response simulation of three controllers when a load torque is applied on the motor at rated speed

Fig. 4.4 shows the generated torque of motor under three control algorithms. It is seen that all three approaches have a fast torque dynamic, however the DTC-SVM algorithm has more torque ripple compared to the other two approaches. Table. 4.3, shows the calculated integral time absolute error (ITAE) for the three algorithms. It is seen that the ITAE of ADP is slightly better than those of FOC and DTC-SVM.

The second simulation considers parameter uncertainties in the motor. In order to evaluate the performance of the algorithms in handling uncertainties, the controllers are tuned using nominal

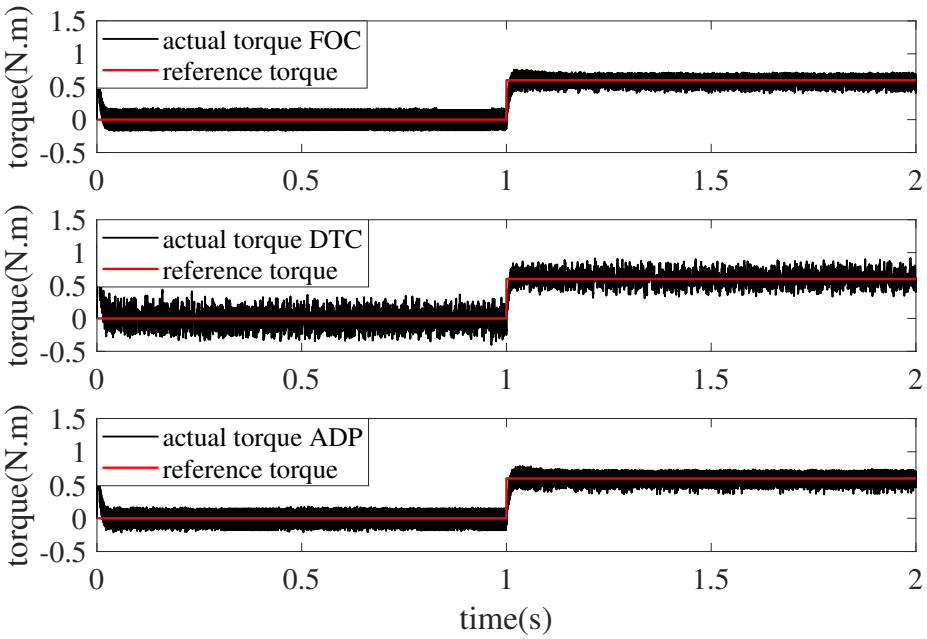


Figure 4.4: Torque response simulation of three controllers when a load torque is applied on the motor at the rated speed of 3000 rpm

values, however in online control, the parameters of the motor with uncertainties are used for simulation. In other words, for instance, in ADP, all the trainings are done with the nominal parameters of Table 4.1, but the motor simulations are done with parameters of Table 4.5. This way the controller has not learned how to react to new parameters. This test can be used to demonstrate the relative robustness of each algorithm. Again the motor response is tested when it is rotating at the rated speed and a load torque of 0.6 $N.m$ is applied at $t = 1s$. It is seen in Fig. 4.5 that before the load is applied, all three controllers can track the desired speed, however when the load is applied at $t = 1s$, only ADP is capable of tracking the reference. A poor performance for DTC-SVM was expected, given its model dependency. However, FOC is not explicitly dependent on the parameters like stator reluctance and resistance as well as permanent magnet flux. Therefore, one may expect an acceptable performance from FOC under such modeling imperfections. But, as see in this figure, FOC also fails, once the load torque is applied. The reason for the poor performance of FOC approach is that, the current loop PI gains are tuned based on the motor nominal parameters. Therefore, when the motor parameters have changed to the parameters in Table.4.5, the PI controller

Table 4.3: ITAE of motor torque when a load torque of 0.6 $N.m$ is applied after one second on the motor at the rated speed of 3000 rpm

Control Algorithm	Torque ITAE
FOC	0.0251
DTC-SVM	0.0287
ADP	0.0245

Table 4.5: Motor Parameters with uncertainties

Parameter	Value
Permanent magnet flux, $\hat{\lambda}_m$	0.012 Wb
Stator resistance, $\hat{R}_s$	5.7 Ω
Stator inductance, $\hat{L}_s$	0.001 H
Rotor moment of inertia, $\hat{J}$	40×10^{-6} $kg.m^2$

of FOC cannot deal with the uncertainties.

In conclusion of the simulation study, one sees that ADP provides significantly better results in dealing with parameter uncertainties. However, It should be noted that the ADP controller is not analytically designed to be robust to parameter uncertainties and it is not claimed here that ADP is robust to structured and unstructured uncertainties. However, because of the feedback nature of the controller and its learning capability, it is seen in simulations that uncertainties are managed to some extent. If the uncertainties are more significant, the desirable performance is not guaranteed.

4.5. Experimental results

In this section, the proposed ADP approach is implemented on a hardware testbed and its performance is compared with FOC and DTC-SVM, experimentally. Fig. 4.6 shows the experimental setup. The PMSM model parameters are the same as Table. 4.1. A BLDC motor is used as a load to apply external step torque on PMSM motor shaft. The Texas Instrument development kit

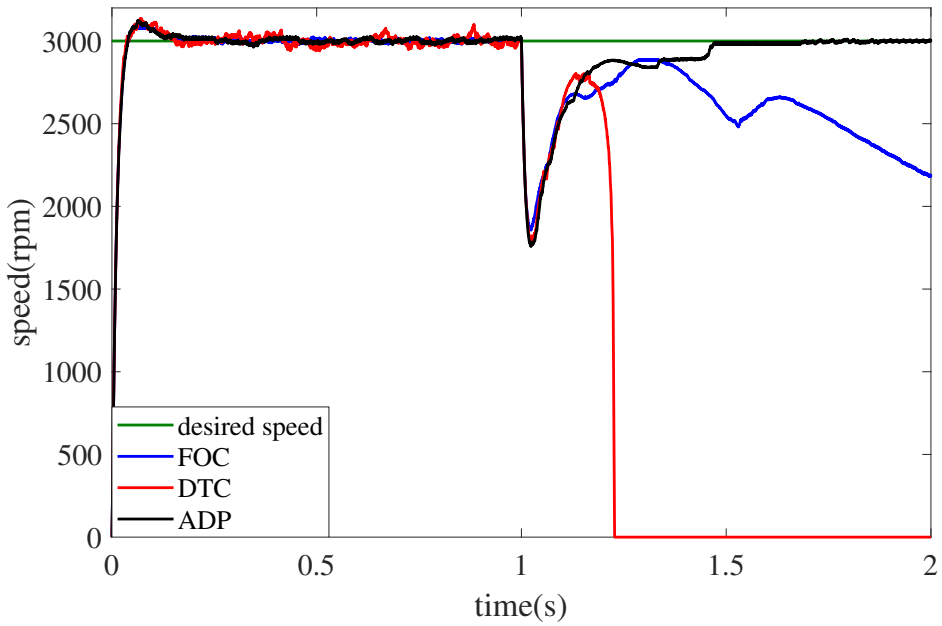


Figure 4.5: Speed response simulation of three controllers when a load torque is applied on the motor at rated speed with parameter uncertainties

TMDSHVMTRINSPIN, with a microcontroller TMS320F28069M is used to implement controllers. This microcontroller has a 90 MHz clock, 16 PWM channels and 16 channels of 12-Bit analog to digital converters (ADC). It is also capable of doing floating point operations very fast, which makes it a suitable processor for this application. The experiments in this section are done under both nominal motor parameters and parameter uncertainties.

4.5.1. Comparative experiment under nominal condition

The same actor weights, which were obtained in simulation are used for the experiments in this section. It should be noted that, in order to have a fair comparison, the PI gains of the speed loop are considered the same for all three algorithms. In the first experiment, the motor is running at $2000rpm$ and a step load of $0.7N.m$ is applied on the motor at $t = 2.3s$. Fig. 4.7, shows the response of each algorithm. Two sections of the figure are magnified to show the transient response and steady state response of the speed. It is observed that ADP and DTC-SVM are faster than FOC in reaching the desired speed after the torque is applied. However, DTC-SVM shows more ripples

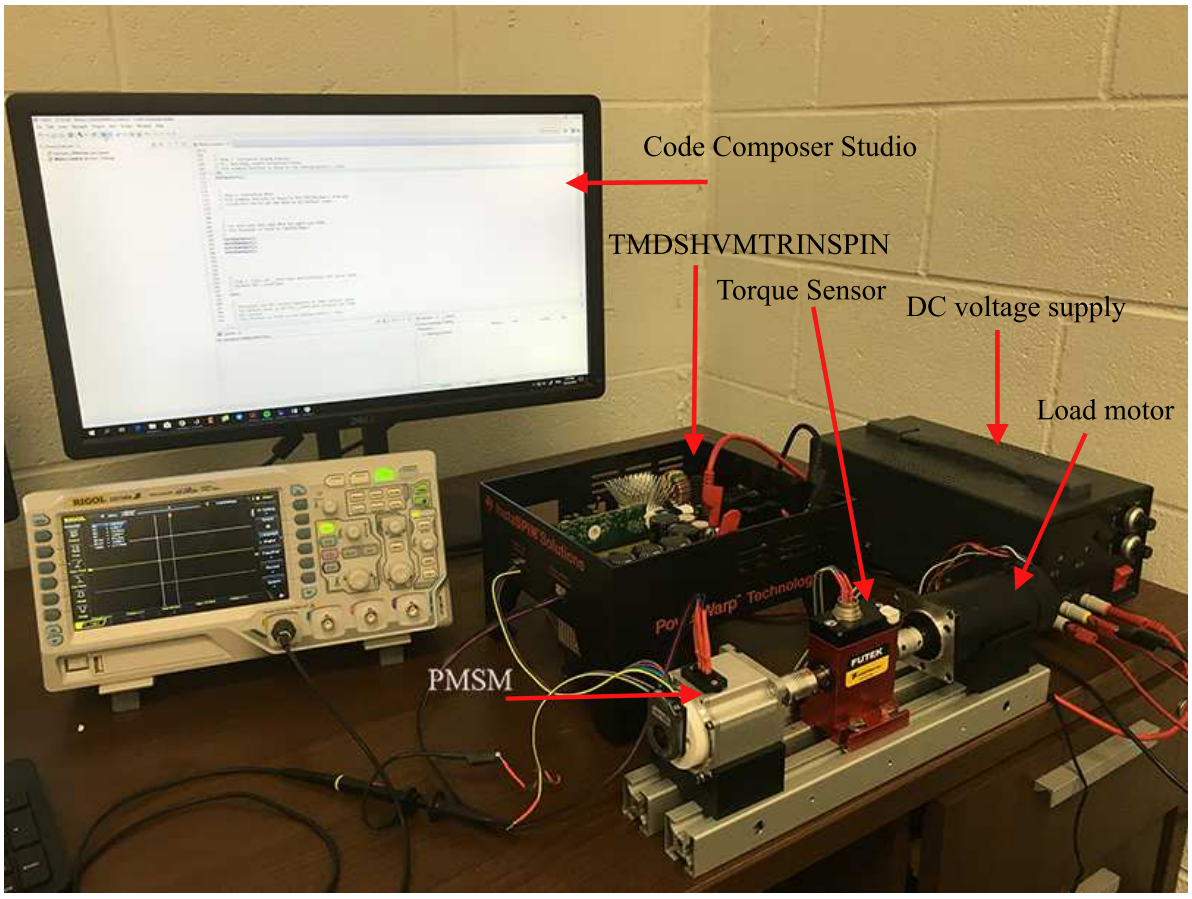


Figure 4.6: PMSM control experimental setup

during steady state.

For the above experiment, the torque is also measured using a torque sensor, namely, FUTEK FSH02564 . Fig. 4.8 shows the generated torque for each algorithm. The vertical dashed lines show the time duration of the transient response of the torque. This time duration for FOC, DTC-SVM, and ADP is 0.4521s, 0.0462s, and 0.039s, respectively. It is seen that the torque response dynamics for FOC is slower than ADP and DTC-SVM.

Table 4.7 shows the ITAE of speed and torque for each algorithms, as points at a slightly better performance for ADP. Therefore, not only the dynamic response of ADP is fast, but also its torque ripples are comparable with FOC. So both transient and steady state responses are satisfactory with ADP. The cost function, defined in (4.6), with Q and R as (4.19) and (4.20) is calculated for FOC, DTC-SVM and ADP as 0.54, 0.56, 0.51, respectively. Although the costs are close, ADP has the

lowest cost among the all three approaches.

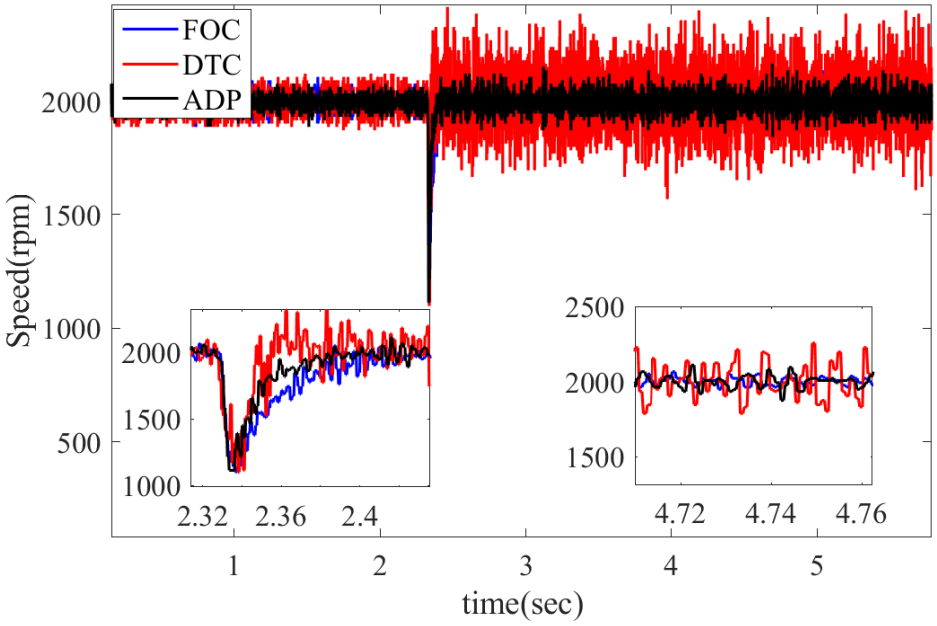


Figure 4.7: Speed response experiment of three controllers when a load of 0.7 $N.m$ is applied at 2000 rpm

Table 4.7: ITAE of motor torque and speed when a load torque of 0.7 $N.m$ is applied at 2000 rpm

Control Algorithm	Speed ITAE	Torque ITAE
FOC	146.6808	1.1994
DTC-SVM	268.0128	1.2143
ADP	131.0423	1.1988

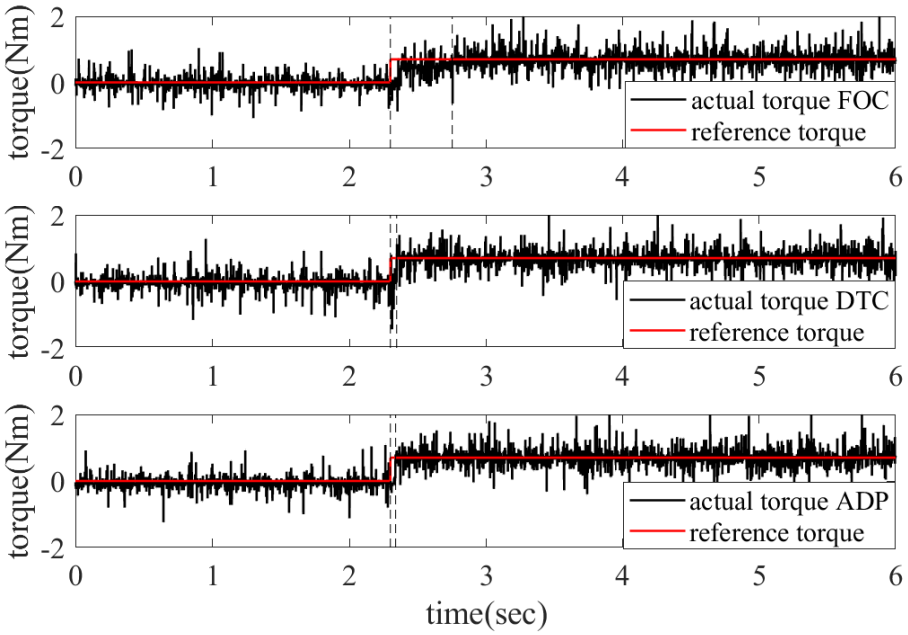


Figure 4.8: Torque response experiment of three controllers when a load of 0.7 $N.m$ is applied at 2000 rpm

In the next experiment, the capability of each algorithm in tracking a desired speed under full load of 0.7 $N.m$ is analyzed. The desired speed is given by

$$\begin{cases} \omega_{ref} = 500rpm & \text{if } 0s \leq t \leq 0.5s \\ \omega_{ref} = 1000rpm & \text{if } 0.5s \leq t \leq 3s \\ \omega_{ref} = 2000rpm & \text{if } 3s \leq t \leq 6s \end{cases} \tag{4.29}$$

As seen in Fig. 4.11, DTC-SVM shows high ripples in tracking the desired response. Both FOC and ADP are able to track the desired reference.

4.5.2. Comparative experiment with parameter variations

In order to show the response of each control algorithm in the presence of parameter uncertainties, it is assumed that the information about the motor parameters are incorrect. These incorrect parameters are considered as $R_s = 3.6\Omega$, $L_s = 0.001H$, and $\lambda_m = 0.005Wb$. Therefore, controllers

are designed based on the incorrect parameter values. However, the actual values are used for the experiment. For ADP, these perturbed parameters are used to obtain the new actor weights. Fig. 4.9 shows the speed response of each algorithm when a load of 0.7 $N.m$ is applied at $t = 3.2s$. It is seen that because of the parameter uncertainties, DTC-SVM can not track the desired speed either before or after the load is applied. This was expected since DTC-SVM is dependent on motor parameters. However, ADP and FOC still show good performances.

The measured torque is also shown in Fig. 4.10. The vertical dashed lines show the time duration of the transient response of the torque. This time duration for FOC, DTC-SVM, and ADP is 0.3621s, 0.3702s, and 0.012s, respectively. It is observed that, the torque dynamics of DTC-SVM, which was fast with nominal parameters, is now slower under parameter uncertainties. However, the time duration for FOC and ADP has not changed significantly.

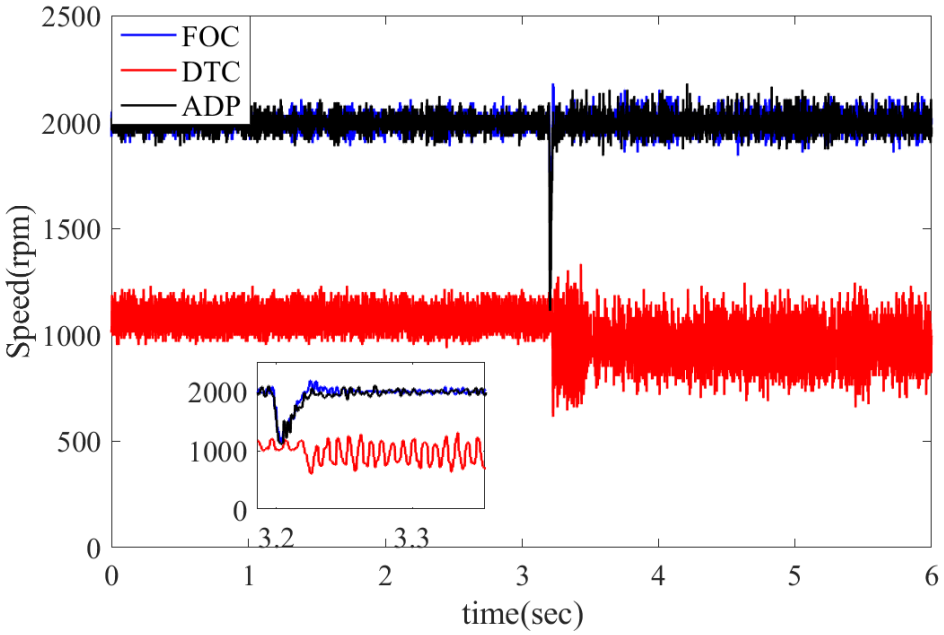


Figure 4.9: Speed response experiment of three controllers under parameter uncertainties when a step load of 0.7 $N.m$ is applied at 2000 rpm

Table 4.9 shows the ITAE of speed and torque for each algorithms. The cost function for FOC, DTC-SVM and ADP as 0.5418, 0.5482, 0.5413, respectively.

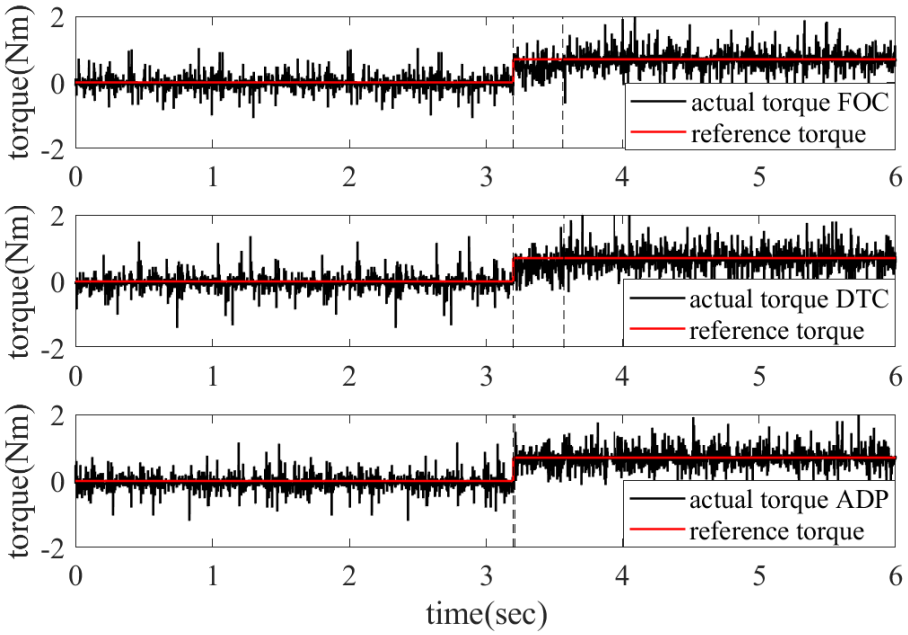


Figure 4.10: torque response experiment of three controllers under parameter uncertainties when a step load of 0.7 $N.m$ is applied at 2000 rpm

The speed tracking of three algorithms when the motor is under the load of 0.7 $N.m$ is also investigated. The reference speed is as (4.29). Fig. 4.12 shows superior performance of ADP compared to FOC and DTC-SVM under parameter uncertainties. It is observed that neither FOC nor DTC-SVM can track the desired speed in this case.

All in all, the experiments have also verified the results demonstrated in the simulations. These experiments justifies the relative robustness of ADP to parameter uncertainties. However, as stated before, if the parameter uncertainties are more significant, there is no guarantee to achieve a desired performance.

4.6. Conclusion

An ADP-based control approach is proposed in this section to achieve fast and accurate torque control in a PMSM. The critic and actor weights are trained once offline and the calculated optimal actor weights are utilized in online control. The simulations and experimental results show that while the advantages of ADP over popular practices, namely, FOC and DTC-SVM are small in the

Table 4.9: ITAE of motor torque and speed under parameter uncertainties when a load torque of 0.7 $N.m$ is applied at 2000 rpm

Control Algorithm	Speed ITAE	Torque ITAE
FOC	133.5428	1.1994
DTC-SVM	2419	1.3943
ADP	132.0423	1.1988

case of *perfect modeling* of the dynamics, the improvements are significant under the case of having *modeling uncertainties* which is a *reality* in any application.

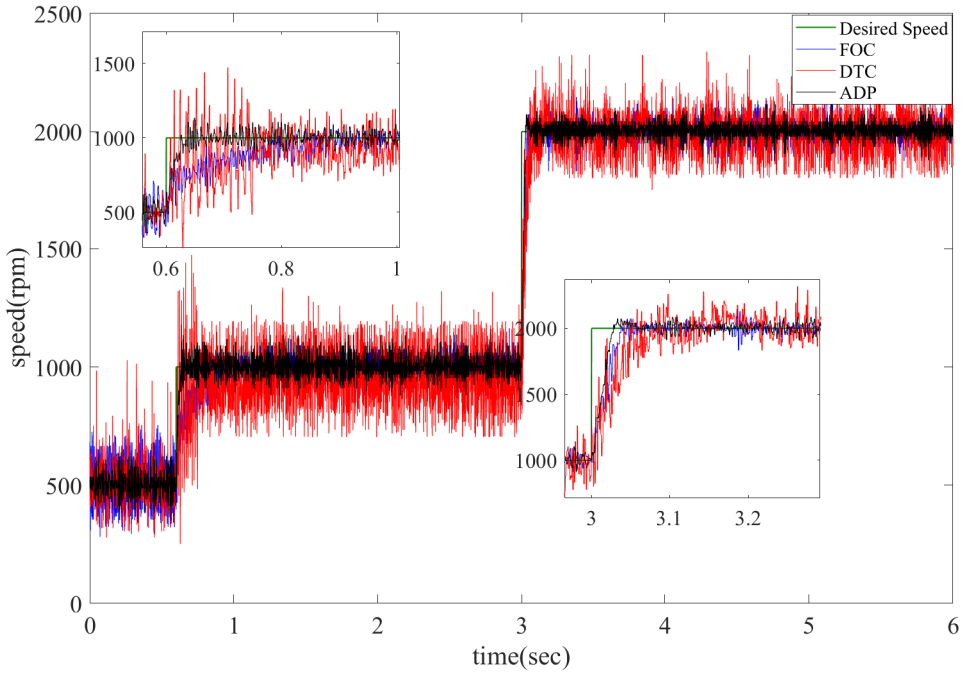


Figure 4.11: Speed response experiment of three controllers when motor is under the load of 0.7 $N.m$

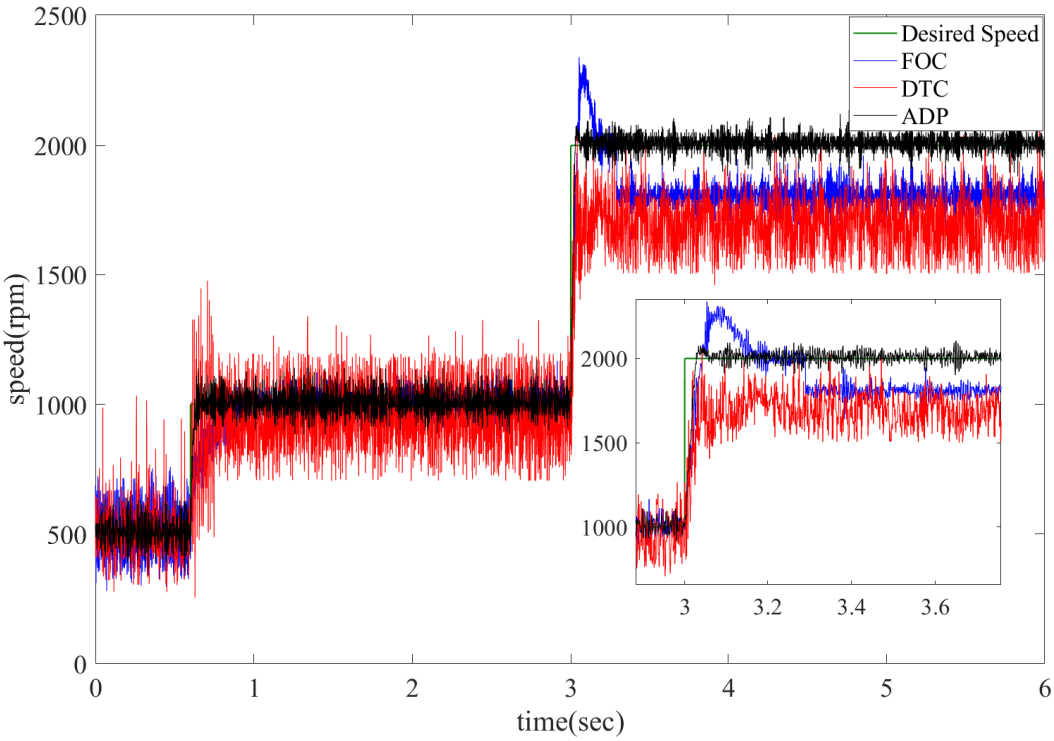


Figure 4.12: Speed response experiment of three controllers under parameter uncertainties when motor is under the load of 0.7 $N.m$

Chapter 5

Conclusions and Future Work

5.1. Conclusions

This work has presented a detailed investigation on the optimal control of single-phase DC-AC inverters and three-phase permanent magnet synchronous motors (PMSMs). An ADP-based controller is presented and it has been implemented for the previously mentioned applications.

For the case of a single-phase inverter, a non-PWM variable-switching-frequency method based on adaptive dynamic programming is proposed to convert DC voltage to AC voltage. The critic network is trained once offline and the converged weights are used in online control to determine the optimal switching schedule. The performance of the controller under different loads is analysed both with simulations and experiments. Finally, the controller is modified and the inverter is upgraded to be used as single phase VFD and the results are shown both numerically and experimentally.

After a brief introduction to the PMSM control problem and explanation of field-oriented control (FOC) and direct torque control (DTC) as two common control approaches for AC machines, the optimal control of a PMSM is presented. An ADP-based control approach is utilized to achieve fast and accurate torque control in a PMSM. The critic and actor weights are trained once offline and the calculated optimal actor weights are utilized in online control. The simulations and experimental results show that while the advantages of ADP over popular practices, namely, FOC and DTC-SVM are small in the case of *perfect modeling* of the dynamics, the improvements are significant under the case of having *modeling uncertainties* which is a *reality* in any application.

5.2. Future Work

Considering the developed work in this thesis, some suggestions can be made for the future work. The first suggestion can be the design and implementation of an ADP-based controller for the

three-phase inverter. One important issue is connecting the inverter to the grid, which can be further studied.

Controlling a PMSM using an ADP-based approach, during field weakening operation can be another topic for further research. This work has only considered the case of normal operation of a PMSM without field weakening. This work can also be extended such that the software identifies the motor parameters, and calculates the optimal weights automatically. Therefore, the user only connects the motor to the board and the software does the rest.

Appendix A

Motor Control Algorithms Using Adaptive Dynamic Programming

Algorithm 1: The pseudocode for the offline training of actor and critic weights

Initialization : Choose the following:
Scaling factors according to (4.26).
$\bar{n}$ random state $\eta^{[q]} \in \Omega, \forall q \in \{1,2,...,\bar{n}\}$
Basis functions $\phi(\eta)$, $\sigma(\eta)$.
Q and R according to (4.19), (4.20).
β_v and β_u as convergence tolerances.

1 Set $i = 0$.
2 Initialize $V^i(\eta^{[q]}) = 0, \forall q \in \{1,2,...,\bar{n}\}$.
3 Find W_c^i from input-target pair $(\eta^{[q]}, V^i(\eta^{[q]}))$.
4 Set $j = 0$.
5 Initialize $u^{i,j}(\eta^{[q]})$ with a random value.
6 Calculate $u^{i,j+1}(\eta^{[q]})$ from (4.23).
7 If $||u^{i,j+1}(\eta^{[q]}) - u^{i,j}(\eta^{[q]})|| < \beta_u$, proceed to the next step, otherwise set $j = j+1$ and go back to step 6.
8 Calculate $V^{i+1}(\eta^{[q]}), \forall q \in \{1,2,...,\bar{n}\}$ from (4.21).
9 If $||V^{i+1}(\eta^{[q]}) - V^i(\eta^{[q]})|| < \beta_v$, proceed to the next step, otherwise go back to step 3.
10 Find W_a^* from input-target pair $(\eta^{[q]}, u^i(\eta^{[q]}))$.

Algorithm 2: The pseudocode for the online implementation of motor control

Initialization : Save the optimal actor weights W_a^*.
Save actor basis function $\sigma(\eta)$.

1 Read motor currents i_a and i_b.
2 Read rotor position θ_m and calculate the rotor speed ω_m.
3 Calculate dq currents from i_a and i_b using (4.1).
4 Pass the error between desired speed and actual speed to a PI controller to get the reference torque.
5 Use (4.26) to normalize the currents, speed and torque.
6 Calculate v_d and v_q using $W_a^*\sigma(\eta)$
7 Feed v_d and v_q to the SVM block to generate the appropriate switching.
8 Go back to step 1.

Appendix B

Torque Sensor Datasheet

FUTEK
ADVANCED SENSOR TECHNOLOGY, INC.

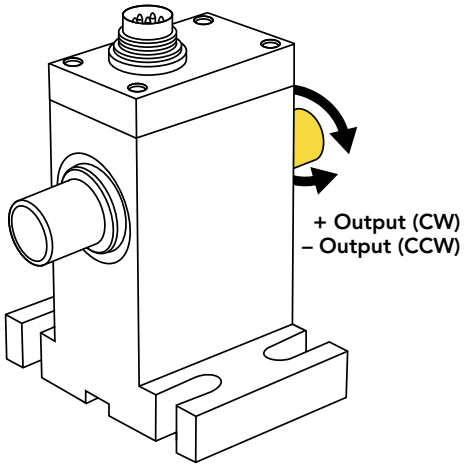
MODEL TRS705

Non-Contact Shaft-to-Shaft Rotary Torque Sensor with Encoder

FEATURES

- Utilizes strain gauge technology
- Angle speed feedback included
- Compact size
- Can operate up to 7000 RPM

Active end



SPECIFICATIONS	
PERFORMANCE	
Nonlinearity	±0.2% of RO
Hysteresis	±0.1% of RO
Nonrepeatability	±0.2% of RO
Rotational Speed	7000 Max
ELECTRICAL	
Rated Output (RO)	±5 VDC
Excitation (VDC or VAC)	11–26 VDC, 1 Watt
Connection	12 pin Binder Series #581 (09-0331-90-12)
MECHANICAL	
Safe Overload	150% of RO
Material	Aluminum (Housing) Steel Alloy (Shaft)
IP Rating	IP40
TEMPERATURE	
Operating Temperature	–13 to 176°F (–25 to 80°C)
Compensated Temperature	41 to 122°F (5 to 50°C)
Temperature Shift Zero	±0.01% of RO/°F (±0.02% of RO/°C)
Temperature Shift Span	±0.01% of RO/°F (±0.02% of RO/°C)
CALIBRATION	
Calibration Test Excitation	12 VDC
Calibration (standard)	Certificate of Conformance
Calibration (available)	5-pt CW & CCW
Shunt Calibration Value	With sensor fully connected apply 11–26 VDC to Pins A & K to generate 5 VDC nom output
ENCODER	
Output	Impulse (TTL)
Pulses per Revolution	2 × 360
Excitation	5 VDC, 40 mA max
Angle 1	Leading Pulse
Angle 2	Trailing Pulse (90°)
CONFORMITY	
RoHS	2014/30/EU
CE	Declaration of Conformity

Model TRS705

TORQUE CONNECTIONS

PIN	COLOR	DESCRIPTION
C	Green	+ Voltage Output
D	White	– Voltage Output
E	Black	Ground
F	Red	Power Supply

ANGLE CONNECTOR CODES

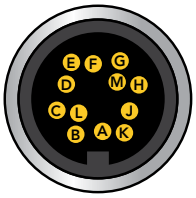
PIN	COLOR	DESCRIPTION
B	Blue	Signal (Angle 1)
E	Black	Ground
G	Brown	Signal (Angle 2)
H	Orange	Power

SHUNT CAL CONNECTOR CODES

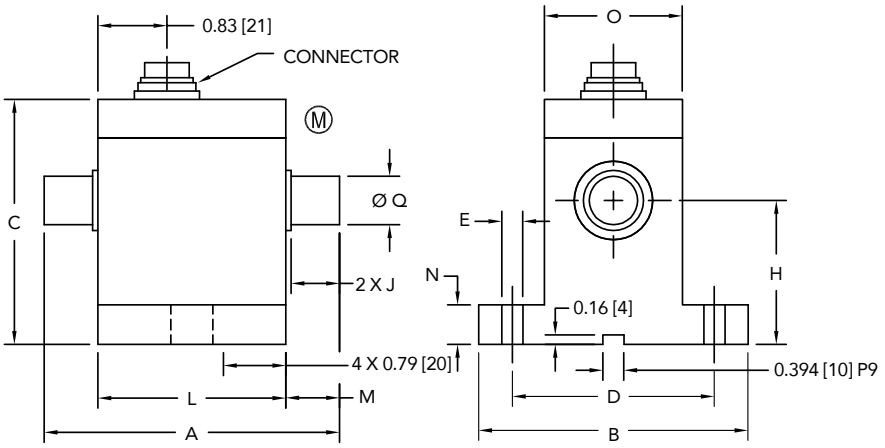
PIN	COLOR	DESCRIPTION
A	Yellow	Ground
K	Purple	Power

SHIELD CONNECTOR CODES

PIN	COLOR	DESCRIPTION
M	Floating	Shield

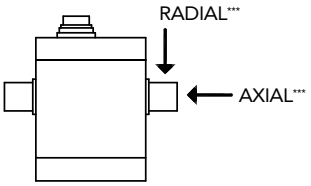


DIMENSIONS inches [mm]



Ⓜ = MEASURING SIDE

* MAXIMUM LOAD ALLOWED. NOT FOR MEASUREMENT.



MASS MOMENT OF INERTIA (kg × cm²)

ITEM #	Measuring End	Drive End
FSH02562	0.0146	0.0147
FSH02563	0.0146	0.0147
FSH02564	0.015	0.015
FSH02565	0.015	0.015
FSH02566	0.062	0.061
FSH02567	0.064	0.063

CAPACITIES

ITEM #	Nm [in-lb]	Ø Q	A	B	C	D	E	H	J	L	M	N	O	*** Max Axial Force lb [N]	*** Max Radial Force lb [N]	Weight lb [kg]	Torsional Stiffness Nm/rad
FSH02562	1 [9]	0.394 [10] g6	3.54 [90]	2.28 [58]	3.27 [83]	1.77 [45]	0.28 [7]	1.77 [45]	0.59 [15]	2.28 [58]	0.63 [16]	0.47 [12]	1.10 [28]	4.5 [20]	3 [15]	1.1 [0.50]	317
FSH02563	2 [18]													11 [50]	5 [25]		317
FSH02564	5 [44]													22 [100]	11 [50]		855
FSH02565	10 [89]													33 [150]	11 [50]		855
FSH02566	20 [177]	0.669 [17] g6	4.17 [106]	3.35 [85]	4.02 [102]	2.36 [60]	0.35 [9]	2.48 [63]	0.87 [22]	2.32 [59]	0.91 [23]	0.59 [15]	1.50 [38]	33 [150]	33 [150]	2.2 [1.00]	4580
FSH02567	50 [443]													45 [200]	33 [150]		8190

CE RoHS ANSI Z540-1 ISO 17025 ISO 9001 ISO 13485 U.S. Manufacturer

BIBLIOGRAPHY

[1] ABRISHAMIFAR, A., AHMAD, A., AND MOHAMADIAN, M. Fixed switching frequency sliding mode control for single-phase unipolar inverters. *IEEE Transactions on Power Electronics 27*, 5 (2012), 2507–2514.

[2] ATAKHIABANI. Pmsm-torque-control-with-adp. https://github.com/AtaKhiabani/PMSM-torque-control-with-ADP.git/, 2019.

[3] ATAKHIABANI. Single-phase-inverter. https://github.com/AtaKhiabani/Single-phase-inverter.git/, 2019.

[4] BERTSEKAS, D. P., AND TSITSIKLIS, J. N. *Neuro-dynamic programming*, vol. 5. Athena Scientific Belmont, MA, 1996.

[5] BLASCHKE, F. The principle of field orientation as applied to the new transvector closed-loop system for rotating-field machines. *Siemens review 34*, 3 (1972), 217–220.

[6] BOZORGI, A. M., FARASAT, M., AND JAFARISHIADEH, S. Model predictive current control of surface-mounted permanent magnet synchronous motor with low torque and current ripple. *IET Power Electronics 10*, 10 (2017), 1120–1128.

[7] CARBALLO, R. E., BOTTERÓN, F., OGGIER, G. G., AND GARCÍA, G. O. Design approach of discrete-time resonant controllers for uninterruptible power supply applications through frequency response analysis. *IET Power Electronics 9*, 15 (2016), 2871–2879.

[8] CHAOUI, H., AND SICARD, P. Adaptive fuzzy logic control of permanent magnet synchronous machines with nonlinear friction. *IEEE Transactions on Industrial Electronics 59*, 2 (2011), 1123–1133.

[9] CHATTOPADHYAY, S., MITRA, M., AND SENGUPTA, S. *Clarke and Park Transform*. Springer Netherlands, Dordrecht, 2011, pp. 89–96.

[10] CHEN, W., ZENG, S., ZHANG, G., SHI, T., AND XIA, C. A modified double vectors model predictive torque control of permanent magnet synchronous motor. *IEEE Transactions on Power Electronics* (2019).

[11] CORTAJARENA, J. A., BARAMBONES, O., ALKORTA, P., AND DE MARCOS, J. Sliding mode control of grid-tied single-phase inverter in a photovoltaic mppt application. *Solar Energy 155* (2017), 793–804.

[12] DOMINGUEZ, J. R., NAVARRETE, A., MEZA, M. A., LOUKIANOV, A. G., AND CANEDO, J. Digital sliding-mode sensorless control for surface-mounted pmsm. *IEEE Transactions on Industrial Informatics 10*, 1 (2013), 137–151.

[13] DURANAY, Z. B., AND GULDEMIR, H. Selective harmonic eliminated v/f speed control of single-phase induction motor. *IET Power Electronics 11*, 3 (2017), 477–483.

[14] EI-SOUSY, F. Hybrid h∞ based wavelet-neuralnetwork tracking control for permant-magnet synchronous motor servo drive. *IEEE Trans. On Industrial Electronic 57*, 9 (2010), 3157–3166.

[15] ESTIMA, J. O. *Development and analysis of permanent magnet synchronous motor drives with fully integrated inverter fault-tolerant capabilities*. PhD thesis, 2012.

[16] EVRAN, F. Plug-in repetitive control of single-phase grid-connected inverter for ac module applications. *IET Power Electronics 10*, 1 (2017), 47–58.

[17] FENG, G., LAI, C., AND KAR, N. C. A closed-loop fuzzy-logic-based current controller for pmsm torque ripple minimization using the magnitude of speed harmonic as the feedback control signal. *IEEE Transactions on Industrial Electronics 64*, 4 (2016), 2642–2653.

[18] FUJII, T., AND YOKOYAMA, T. FPGA based deadbeat control with disturbance compensator for single phase pwm inverter. In *Power Electronics Specialists Conference, 2006. PESC'06. 37th IEEE* (2006), IEEE, pp. 1–6.

[19] HALE, M., AND WARDI, Y. Mode scheduling under dwell time constraints in switched-mode systems. In *2014 American Control Conference* (2014), IEEE, pp. 3954–3959.

[20] HASANZADEH, A., EDRINGTON, C., MAGHSOUDLOU, B., FLEMING, F., AND MOKHTARI, H. Multi-loop linear resonant voltage source inverter controller design for distorted loads using the linear quadratic regulator method. *IET Power Electronics 5*, 6 (2012), 841–851.

[21] HE, L., ZHANG, K., XIONG, J., AND FAN, S. A repetitive control scheme for harmonic suppression of circulating current in modular multilevel converters. *IEEE transactions on power electronics 30*, 1 (2015), 471–481.

[22] HEYDARI, A. Revisiting approximate dynamic programming and its convergence. *IEEE transactions on cybernetics 44*, 12 (2014), 2733–2743.

[23] HEYDARI, A. Optimal scheduling for reference tracking or state regulation using reinforcement learning. *Journal of the Franklin Institute 352*, 8 (2015), 3285–3303.

[24] HEYDARI, A. Optimal switching of dc-dc power converters using approximate dynamic programming. *IEEE Transactions on Neural Networks and Learning Systems* (2016).

[25] HEYDARI, A. Optimal switching of dc–dc power converters using approximate dynamic programming. *IEEE transactions on neural networks and learning systems 29*, 3 (2018), 586–596.

[26] HEYDARI, A. Stability analysis of optimal adaptive control under value iteration using a stabilizing initial policy. *IEEE transactions on neural networks and learning systems 29*, 9 (2018), 4522–4527.

[27] HEYDARI, A., AND BALAKRISHNAN, S. N. Optimal switching between autonomous subsystems. *Journal of the Franklin Institute 351*, 5 (2014), 2675–2690.

[28] IEC, I. 62040-3: Uninterruptible power systems (ups)-part 3: Method of specifying the performance and test requirements. *Switzerland: IEC* (2004).

[29] JASZCZOLT, C. Understanding permanent magnet motors. https://www.controleng.com/articles/understanding-permanent-magnet-motors/, 2019.

[30] JUNG, J.-W., LEU, V. Q., DO, T. D., KIM, E.-K., AND CHOI, H. H. Adaptive pid speed control design for permanent magnet synchronous motor drives. *IEEE Transactions on Power Electronics 30*, 2 (2014), 900–908.

[31] KAMALAPURKAR, R., ANDREWS, L., WALTERS, P., AND DIXON, W. E. Model-based reinforcement learning for infinite-horizon approximate optimal tracking. *IEEE transactions on neural networks and learning systems 28*, 3 (2017), 753–758.

[32] KAWAMURA, A., CHUARAYAPRATIP, R., AND HANEYOSHI, T. Deadbeat control of pwm inverter with modified pulse patterns for uninterruptible power supply. *IEEE Transactions on Industrial Electronics 35*, 2 (1988), 295–300.

[33] KENNY, B. H., AND LORENZ, R. D. Stator-and rotor-flux-based deadbeat direct torque control of induction machines. *IEEE Transactions on industry applications 39*, 4 (2003), 1093–1101.

[34] KHAJEHODDIN, S. A., KARIMI-GHARTEMANI, M., JAIN, P. K., AND BAKHSHAI, A. A resonant controller with high structural robustness for fixed-point digital implementations. *IEEE Transactions on Power Electronics 27*, 7 (2012), 3352–3362.

[35] KHIABANI, A. G., AND HEYDARI, A. Optimal switching of voltage source inverters using approximate dynamic programming. In *ASME 2018 Dynamic Systems and Control Conference* (2018), American Society of Mechanical Engineers, pp. V001T01A006–V001T01A006.

[36] KHIABANI, A. G., AND HEYDARI, A. Design and implementation of an optimal switching controller for uninterruptible power supply inverters using adaptive dynamic programming. *IET Power Electronics 12*, 12 (2019), 3068–3076.

[37] KIM, S., AND AHN, C. K. Offset-free proportional-type self-tuning speed controller for permanent magnet synchronous motors. *IEEE Transactions on Industrial Electronics* (2018), 1–1.

[38] KIRK, D. E. *Optimal control theory: an introduction*. Courier Corporation, 2012.

[39] KIUMARSI, B., VAMVOUDAKIS, K. G., MODARES, H., AND LEWIS, F. L. Optimal and autonomous control using reinforcement learning: A survey. *IEEE transactions on neural networks and learning systems 29*, 6 (2018), 2042–2062.

[40] KOMURCUGIL, H., BAYHAN, S., AND ABU-RUB, H. Variable and fixed switching frequency based hcc methods for grid-connected vsi with active damping and zero steady-state error. *IEEE Transactions on Industrial Electronics* (2017).

[41] KOURO, S., CORTÉS, P., VARGAS, R., AMMANN, U., AND RODRÍGUEZ, J. Model predictive control—a simple and powerful method to control power converters. *IEEE Transactions on Industrial Electronics 56*, 6 (2009), 1826–1838.

[42] KRÖSE, B., KROSE, B., VAN DER SMAGT, P., AND SMAGT, P. An introduction to neural networks.

[43] KUKRER, O., KOMURCUGIL, H., AND DOGANALP, A. A three-level hysteresis function approach to the sliding-mode control of single-phase ups inverters. *IEEE Transactions on Industrial Electronics 56*, 9 (2009), 3477–3486.

[44] LAI, Y.-S., AND CHEN, J.-H. A new approach to direct torque control of induction motor drives for constant inverter switching frequency and torque ripple reduction. *IEEE Transactions on Energy Conversion 16*, 3 (2001), 220–227.

[45] LASCU, C., AND TRZYNADLOWSKI, A. M. Combining the principles of sliding mode, direct torque control, and space-vector modulation in a high-performance sensorless ac drive. *IEEE Transactions on industry applications 40*, 1 (2004), 170–177.

[46] LEWIS, F. L., AND VRABIE, D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE circuits and systems magazine 9*, 3 (2009).

[47] LEWIS, F. L., VRABIE, D., AND SYRMOS, V. L. *Optimal control*. John Wiley & Sons, 2012.

[48] LEWIS, F. L., VRABIE, D., AND VAMVOUDAKIS, K. G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine 32*, 6 (2012), 76–105.

[49] LI, B., HUANG, S., CHEN, X., AND WAN, S. Enhanced dq current control for single-phase voltage-source inverters. *IET Power Electronics* (2018).

[50] LI, S., WON, H., FU, X., FAIRBANK, M., WUNSCH, D. C., AND ALONSO, E. Neural-network vector controller for permanent-magnet synchronous motor drives: Simulated and hardware-validated results. *IEEE transactions on cybernetics* (2019).

[51] LIM, C.-S., LEVI, E., JONES, M., RAHIM, N. A., AND HEW, W.-P. A comparative study of synchronous current control schemes based on fcs-mpc and pi-pwm for a two-motor three-phase drive. *IEEE Transactions on Industrial Electronics 61*, 8 (2013), 3867–3878.

[52] LIU, D., WANG, D., ZHAO, D., WEI, Q., AND JIN, N. Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming. *IEEE Transactions on Automation Science and Engineering 9*, 3 (2012), 628–634.

[53] LIU, J., LI, H., AND DENG, Y. Torque ripple minimization of pmsm based on robust ilc via adaptive sliding mode control. *IEEE Transactions on Power Electronics 33*, 4 (2017), 3655–3671.

[54] LIU, M., CHAN, K. W., HU, J., XU, W., AND RODRIGUEZ, J. Model predictive direct speed control with torque oscillation reduction for pmsm drives. *IEEE Transactions on Industrial Informatics* (2019).

[55] LOXTON, R. C., TEO, K. L., REHBOCK, V., AND LING, W. Optimal switching instants for a switched-capacitor dc/dc power converter. *Automatica 45*, 4 (2009), 973–980.

[56] MAO, H., YANG, X., CHEN, Z., AND WANG, Z. A hysteresis current controller for single-phase three-level voltage source inverters. *IEEE Transactions on Power Electronics 27*, 7 (2012), 3330–3339.

[57] MELFI, M. J., EVON, S., AND MCELVEEN, R. Induction versus permanent magnet motors. *IEEE Industry Applications Magazine 15*, 6 (2009), 28–35.

[58] MENDEL, J., AND MCLAREN, R. 8 reinforcement-learning control and pattern recognition systems. In *Mathematics in Science and Engineering*, vol. 66. Elsevier, 1970, pp. 287–318.

[59] MOHSENI, M., AND ISLAM, S. M. A new vector-based hysteresis current control scheme for three-phase pwm voltage-source inverters. *IEEE transactions on power electronics 25*, 9 (2010), 2299–2309.

[60] MONFARED, M. A simplified control strategy for single-phase ups inverters. *Bulletin of the Polish Academy of Sciences Technical Sciences 62*, 2 (2014), 367–373.

[61] NGUYEN, A. T., RAFAQ, M. S., CHOI, H. H., AND JUNG, J.-W. A model reference adaptive control based speed controller for a surface-mounted permanent magnet synchronous motor drive. *IEEE Transactions on Industrial Electronics 65*, 12 (2018), 9399–9409.

[62] NIU, F., WANG, B., BABEL, A. S., LI, K., AND STRANGAS, E. G. Comparative evaluation of direct torque control strategies for permanent magnet synchronous machines. *IEEE Transactions on Power Electronics 31*, 2 (2015), 1408–1424.

[63] PANAPAKIDIS, I. P., SARAFIANOS, D. N., AND ALEXIADIS, M. C. Comparative analysis of different grid-independent hybrid power generation systems for a residential load. *Renewable and Sustainable Energy Reviews 16*, 1 (2012), 551–563.

[64] PINILLA, M., AND MARTINEZ, S. Selection of main design variables for low-speed permanent magnet machines devoted to renewable energy conversion. *IEEE Transactions on Energy Conversion 26*, 3 (2011), 940–945.

[65] PREINDL, M., AND BOLOGNANI, S. Model predictive direct torque control with finite control set for pmsm drive systems, part 1: Maximum torque per ampere operation. *IEEE Transactions on Industrial Informatics 9*, 4 (2013), 1912–1921.

[66] QIN, C., ZHANG, H., AND LUO, Y. Optimal tracking control of a class of nonlinear discrete-time switched systems using adaptive dynamic programming. *Neural Computing and Applications 24*, 3-4 (2014), 531–538.

[67] RAZI, R., KARBASFOROOSHAN, M.-S., AND MONFARED, M. Multi-loop control of ups inverter with a plug-in odd-harmonic repetitive controller. *ISA transactions 67* (2017), 496–506.

[68] RAZZAGHI, P., AL KHATIB, E., AND HURMUZLU, Y. Nonlinear dynamics and control of an inertially actuated jumper robot. *Nonlinear Dynamics 97*, 1 (2019), 161–176.

[69] RYMARSKI, Z., AND BERNACKI, K. Different approaches to modelling single-phase voltage source inverters for uninterruptible power supply systems. *IET Power Electronics 9*, 7 (2016), 1513–1520.

[70] SANDRE-HERNANDEZ, O., DE JESUS RANGEL-MAGDALENO, J., AND MORALES-CAPORAL, R. Modified model predictive torque control for a pmsm-drive with torque ripple minimisation. *IET Power Electronics 12*, 5 (2019), 1033–1042.

[71] SARDARMEHNI, T., AND HEYDARI, A. Suboptimal scheduling in switched systems with continuous-time dynamics: A least squares approach. *IEEE transactions on neural networks and learning systems 29*, 6 (2018), 2167–2178.

[72] SARDARMEHNI, T., AND HEYDARI, A. Sub-optimal switching in anti-lock brake systems using approximate dynamic programming. *IET Control Theory & Applications 13*, 9 (2019), 1413–1424.

[73] SOLBAKKEN, Y. Vector control for dummies. https://www.switchcraft.org/learning/2016/12/16/vector-control-for-dummies/, 2017.

[74] UFNALSKI, B., KASZEWSKI, A., AND GRZESIAK, L. M. Particle swarm optimization of the multioscillatory lqr for a three-phase four-wire voltage-source inverter with an output filter. *IEEE Transactions on Industrial Electronics 62*, 1 (2015), 484–493.

[75] VAEZ-ZADEH, S. *Control of permanent magnet synchronous motors*. Oxford University Press, 2018.

[76] VAEZ-ZADEH, S., AND ZAHEDI, B. Modeling and analysis of variable speed single phase induction motors with iron loss. *Energy Conversion and Management 50*, 11 (2009), 2747–2753.

[77] VAFAIE, M. H., DEHKORDI, B. M., MOALLEM, P., AND KIYOUMARSI, A. A new predictive direct torque control method for improving both steady-state and transient-state operations of the pmsm. *IEEE Transactions on power electronics 31*, 5 (2015), 3738–3753.

[78] VAMVOUDAKIS, K. G., AND HESPANHA, J. P. Online optimal switching of single phase dc/ac inverters using partial information. In *2014 American Control Conference* (2014), IEEE, pp. 2624–2630.

[79] VAMVOUDAKIS, K. G., AND HESPANHA, J. P. Online optimal operation of parallel voltage source inverters using partial information. *IEEE Transactions on Industrial Electronics* (2016).

[80] VU, P., NGUYEN, Q., TRAN, M., TODESCHINI, G., AND SANTOSO, S. Adaptive backstepping approach for dc-side controllers of z-source inverters in grid-tied pv system applications. *IET Power Electronics 11*, 14 (2018), 2346–2354.

[81] WANG, Q., YU, H., WANG, M., AND QI, X. A novel adaptive neuro-control approach for permanent magnet synchronous motor speed control. *Energies 11*, 9 (2018), 2355.

[82] XU, S., WANG, J., AND XU, J. A current decoupling parallel control strategy of single-phase inverter with voltage and current dual closed-loop feedback. *IEEE Transactions on Industrial Electronics 60*, 4 (2013), 1306–1313.

[83] XU, Y., DING, X., WANG, J., AND WANG, C. Robust three-vector-based low-complexity model predictive current control with supertwisting-algorithm-based second-order sliding-mode observer for permanent magnet synchronous motor. *IET Power Electronics 12*, 11 (2019), 2895–2903.

[84] XU, Y., WANG, M., ZHANG, W., AND ZOU, J. Sliding mode observer for sensorless control of surface permanent magnet synchronous motor equipped with lc filter. *IET Power Electronics 12*, 4 (2018), 686–692.

[85] YANG, S., LEI, Q., PENG, F. Z., AND QIAN, Z. A robust control scheme for grid-connected voltage-source inverters. *IEEE Transactions on Industrial Electronics 58*, 1 (2011), 202–212.

[86] YI, Y., VILATHGAMUWA, D. M., AND RAHMAN, M. A. Implementation of an artificial-neural-network-based real-time adaptive controller for an interior permanent-magnet motor drive. *IEEE transactions on industry applications 39*, 1 (2003), 96–104.

[87] ZHANG, H., LIU, D., LUO, Y., AND WANG, D. *Adaptive dynamic programming for control: algorithms and stability*. Springer Science & Business Media, 2012.

[88] ZHANG, H., WEI, Q., AND LUO, Y. A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy hdp iteration algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38*, 4 (2008), 937–942.

[89] ZHANG, X., WANG, Y., YU, C., GUO, L., AND CAO, R. Hysteresis model predictive control for high-power grid-connected inverters with output lcl filter. *IEEE Transactions on Industrial Electronics 63*, 1 (2016), 246–256.

[90] ZHANG, X., ZHANG, L., AND ZHANG, Y. Model predictive current control for pmsm drives with parameter robustness improvement. *IEEE Transactions on Power Electronics 34*, 2 (Feb 2019), 1645–1657.

[91] ZHONG, L., RAHMAN, M. F., HU, W. Y., AND LIM, K. Analysis of direct torque control in permanent magnet synchronous motor drives. *IEEE Transactions on Power Electronics 12*, 3 (1997), 528–536.

[92] ZHOU, W., LIU, H., HE, H., YI, J., AND LI, T. Neuro-optimal tracking control for continuous stirred tank reactor with input constraints. *IEEE Transactions on Industrial Informatics 15*, 8 (2019), 4516–4524.

[93] ZHOU, Z., XIA, C., YAN, Y., WANG, Z., AND SHI, T. Disturbances attenuation of permanent magnet synchronous motor drives using cascaded predictive-integral-resonant controllers. *IEEE Transactions on Power Electronics 33*, 2 (Feb 2018), 1514–1527.