POSITION CONTROL OF A PERMANENT MAGNET DC MOTOR BY MODEL REFERENCE ADAPTIVE CONTROL

by

Fedai YENİCİ

July 2006

POSITION CONTROL OF A PERMANENT MAGNET DC MOTOR BY MODEL REFERENCE ADAPTIVE CONTROL

by

Fedai YENİCİ

A thesis submitted to

the Graduate Institute of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electronics Engineering

July 2006
Istanbul, Turkey

APPROVAL PAGE

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Muhammet KÖKSAL
Chairperson
Electronics Engineering Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Muhammet KÖKSAL
Supervisor

Examining Committee Members

Prof. Dr. Muhammet KÖKSAL .....

Assoc. Prof. Dr. Erkan İMAL .....

Assist. Prof. Dr. Atakan KURT .....

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering.

Assist. Prof. Dr. Nurullah ASLAN
Director

Date
July 2006

POSITION CONTROL OF A PERMANENT MAGNET DC MOTOR BY MODEL REFERENCE ADAPTIVE CONTROL

Fedai YENİCİ

M. S. Thesis - Electronics Engineering

July 2006

Supervisor: Prof. Dr. Muhammet KÖKSAL

ABSTRACT

Model reference adaptive control is one of the various techniques of solving the control problem when the parameters of the controlled process are poorly known or vary during normal operation. To understand the dynamic behavior of a dc motor it is required to know its parameters. Armature inductance, armature resistance, inertia of the rotor, motor constants and friction coefficient are the main parameters of a dc motor. To identify all these parameters, some experiments should be performed. However, motor parameters change during the operation according to several conditions. Therefore, the performance of the controller, which has been designed considering constant motor parameters, becomes poorer due to parameter variations. For this reason, a model reference adaptive control method is proposed to control the position of a dc motor without requiring any fixed motor parameter. Experimental results show how well this method controls the position of the motor.

Keywords: Model Reference Adaptive Control, Lyapunov's Direct Method, Dc Motor.

MODEL REFERANS UYARLAMALI KONTROL İLE KUTUPLARI SABİT MIKNATISLI BİR DC MOTORUN KONUM KONTROLU

Fedai YENİCİ

Yüksek Lisans Tezi – Elektronik Mühendisliği

Temmuz 2006

Tez Yöneticisi: Prof. Dr. Muhammet KÖKSAL

ÖZ

Prosesin parametrelerinin tam olarak bilinmemesi veya parametrelerin çalışma esnasında değişmesi durumunda kontrol probleminin çözümünde kullanılabilecek çeşitli tekniklerden biri de model referans uyarlamalı denetimdir. Doğru akım motorunun dinamiğinin anlaşılması için motor parametrelerinin bilinmesi gerekmektedir. Bu parametrelerden başlıcaları armatür endüktansı, armatür direnci, rotorun dönel eylemsizliği, motor sabitleri ve rotor yatağındaki sürtünmedir. Tüm bu parametrelerin belirlenmesi için bazı deneylerin yapılması gerekir. Ancak motor parametreleri, motorun çalışması sırasındaki koşullara bağlı olarak değişmektedir. Dolayısıyla, parametrelerin değişmediği varsayılarak tasarlanmış olan denetleyicinin performansı, çalışma sırasındaki değişimlerden dolayı düşer. Bu nedenden dolayı bu çalışmada, sabit motor parametrelerine gerek kalmadan, bir dc motorun konumunun kontrol edildiği bir uyarlamalı denetim yöntemi sunulmaktadır. Deney sonuçları motor konumunun önerilen yöntem ile denetlenebildiğini göstermiştir.

Anahtar Kelimeler: Model Referans Uyarlamalı Kontrol, Lyapunov'un direkt metodu, Dc Motor.

ACKNOWLEDGEMENT

I would like to thank my advisor, Prof. Dr. Muhammet Köksal, for his knowledge and support in helping me throughout my research. I appreciate his enlightening guidance and advice to me in completing this study. Especially his serious attitude on research and his pursuit for the perfect work will help me in a long run.

I also sincerely thank my committee member, Assoc. Prof. Dr. Erkan İmal, for his patience and kind support for completing this thesis. Their lectures helped me fulfilling this research.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF SYMBOLS AND ABBREVIATIONS

SYMBOL

Δ	Unmodeled plant phenomena
P (θ^*)	Plant model which is parameterized with respect to some unknown parameter vector θ^*
θ^*	Coefficients of the numerator and denominator of the plant transfer function
θ(t)	Estimate of θ^* at each time t
$\hat{P}(\theta(t))$	Estimated plant model
θ_C(t)	Controller parameter
C (θ_C)	Control law
W_M(s)	Transfer function of the reference model
r	Reference signal
u	The input of plant
g	Feed forward gain of reference signal
g^*	Optimal values of feed forward gain of reference signal
$\boldsymbol{A}$	Coefficient matrix of $\boldsymbol{x}$
$\boldsymbol{A_m}$	Coefficient matrix of $\boldsymbol{z}$
$\boldsymbol{b}$	Coefficient matrix of u
$\boldsymbol{b_m}$	Coefficient matrix of r
$\boldsymbol{F}$= $[F_1\ F_2.....F_n]^T$	Feedback gains of plant's states
$\boldsymbol{F}^*$	Optimal values of feedback gains of plant's states
$\boldsymbol{x}$	Controllable canonical form of plant states
$\boldsymbol{z}$	Controllable canonical form of model's states
$\boldsymbol{e}$	Error signal between model and plant states
α	Adaptive gain

$\boldsymbol{s}^T$	Abbreviated form of $\boldsymbol{b}_m^T \boldsymbol{P}$
T	Adaptation period or simulation step size
T_{pi}	Time constants of plant
T_{mi}	Time constants of reference model
T_c	Computation time
f	Adaptation frequency (inverse of T)
$\boldsymbol{P}$	Positive definite matrices for Lyapunov equation
$\boldsymbol{Q}$	Positive definite matrices for Lyapunov equation
ξ	Damping ratio
w_n	Natural undamped frequency
K	Numerator of plant's transfer function
K^*	Numerator of reference model's transfer function
a_j	Denominator coefficients of plant's transfer function
a_j^*	Denominator coefficients of reference model's transfer function
$\boldsymbol{\Phi_F}$	Difference between $\boldsymbol{F}$ and $\boldsymbol{F}^*$
Φ_g	Difference between g^* and g
$\boldsymbol{V}$	Lyapunov function
$i_a(t)$	armature current
R_a	armature resistance
$e_b(t)$	back electromotive force (emf)
$T_L(t)$	load torque
$T_m(t)$	motor torque
$\theta_m(t)$	rotor displacement
K_i	torque constant
K_b	back emf constant
L_a	armature inductance

$e_a(t)$	applied voltage
Φ	magnetic flux in the air gap
$w_m(t)$	rotor angular velocity
$J_m(t)$	rotor inertia
B_m	viscous- friction coefficient
t_d	the delay time
t_r	the rise time
t_s	the settling time

ABBREVIATION

DC	Direct Current
DSP	Digital Signal Processing
DMRAC	Decentralized Model Reference Adaptive Control
FDD	Fast Detection and Diagnosis
HFG	High Frequency Gain
I/O	Input/Output
LOS	Line Of Sight
LTI	Linear Time Invariant.
MRAC	Model Reference Adaptive Control
MRC	Model Reference Control
MIMO	Multi Input Multi Ouput
PM	Permanent Magnet
SISO	Single Input Single Output
SOC	Self Organizing Control
SPR	Strictly Positive Real

CHAPTER I

INTRODUCTION

In spoken language, “to adapt” shows to change a behavior to become similar to new circumstances. Instinctively, an adaptive controller is so a controller that can modify its behavior in response to changes in the dynamics of the process and character of the disturbances. Since ordinary feedback also tries to reduce the effects of disturbances and plant uncertainty, the question of the distinction between feedback control and adaptive control without delay arises. Through the years there have been many efforts to explain adaptive control formally. At an early symposium in 1961 a long discussion ended with the following suggestion: “An adaptive system is any physical system that has been designed with an adaptive viewpoint.” A renewed try was made by an IEEE committee in 1973. It submitted a new vocabulary based on concepts like self-organizing control (SOC), and learning control system. However, these tries were not widely accepted. A meaningful definition of adaptive control, which would make it possible to look at a controller hardware and software and decide whether or not it is adaptive, is still lacking. However, there seems to be a general agreement that a constant – gain feedback system is not an adaptive system (Aström, 1995).

Adaptive systems have two advantages according to non adaptive systems. First of all, if plant parameters change during the operation according to the several conditions an adaptive system adjusts itself, and the performance of the plant becomes as desired. In the non adaptive systems, controller is designed using constant plant parameters. When the plant parameters change, performance of the controller decreases.

Second advantage of adaptive systems is the information of plant’s parameters is not required for a controller design. Control can be made partly or without any plant parameters.

Plant parameters must be known in the non adaptive systems. For this reason, research of the plant parameters brings us extra difficulties.

Generally, the aim of adaptive systems is to control the plant with unknown parameters. There are two techniques for this purpose direct and indirect adaptive control respectively. The plant parameters are estimated on-line in indirect adaptive control. Direct adaptive control doesn't need on line parameter estimation. Direct adaptive control is considered in this thesis.

This method requires knowing plant's zeros. Therefore, we consider plants without finite transmission zeros. For this reason, the method is proper for position control of dc motor. In this thesis the position control of a dc motor is considered by using model reference adaptive control (MRAC). Adaptation mechanism which adjusts recursively plant's feed-forward and feedback gains, tries that equalize the coefficients of closed loop plant to model's coefficients. Hence if we choose proper model, plant output converges to the model output with time.

The parameters in a dc motor are armature inductance, armature resistance, rotor inertia, friction coefficient, motor moment, load moment, armature voltage, motor speed, speed coefficient and moment coefficient. These parameters change with external effects and working conditions. These alterations effect motor dynamic. We will investigate the performance of adaptive tracking in spite of parameters variation.

The solution of problem against to the parameters variations is investigated by many scientists. These are summarized below.

(Zeng at all, 1999) controls of flexible spacecraft with using output feedback and variable structure model reference adaptive control theory. For the derivation of control law, it is assumed that the parameters and the structure of the nonlinear functions in the model are unknown. It is shown that in the closed-loop system including the variable structure model reference adaptive control system designed using bounds on uncertain functions, the pitch angle tracks given reference trajectory and the vibration is suppressed.

The speed which is needed for motor driver of induction motor is estimated from motor current without using sensor. The parallel MRAC is used for speed estimation. The error signal between reference model output and adjustable system output is driven to zero through

an adaptive law. The speed of motor is estimated very fast with using large adaptive gains (Kojabadi, 2005), (Park and Kwon, 2004).

(Zhou and Wang, 2005) control of a permanent magnet synchronous motor with using techniques of MRAC and back stepping control. In the controller design, the input output feedback linearization is first of all used to compensate the nonlinearities in the nominal system. Then, adaptive back stepping control approach is adopted in order to derive the control scheme, which is strong to the parameter uncertainties and load torque disturbance.

(Marino at all, 1998) present an adaptive nonlinear control algorithm for current fed induction motors which is adaptive with regards to both load torque and rotor resistance. The eighth order adaptive controller supplies reference signals for stator currents on the basis of: measurements of rotor speed, stator currents and stator voltages; estimates of rotor fluxes, which are the unmeasured state variables; estimates of torque load and rotor resistance which may vary significantly during operations. The dynamic controller assures speed tracking and bounded signals for every initial condition of the motor. When persistency of excitation conditions are satisfied, the rotor flux tracking error goes asymptotically to zero so that motor power efficiency may be enhanced. In addition, in this case, the estimates of rotor fluxes, torque load and rotor resistance tend asymptotically to their true values. Results show that persistency of excitation conditions are satisfied in physical operating conditions and that all estimation errors go quickly to zero so that high tracking performances are obtained both for speed and rotor flux.

(Lee at all, 2000) present and compare possible intelligent control designs for precision motion control applications which are established upon the use of linear actuators. The control of linear motor is realized three different ways which include adaptive control, composite control using a radial-basis function for nonlinear compensation and an iterative learning control. Experimental results show that first two are more successful than last technique.

(Eldeeb and Elmaraghy, 1998) present a new optimal controller designed for rigid-body robots containing motor dynamics. The new optimal adaptive controller developed in this work is established upon feedback linearization, it does not need acceleration feedback, and it does not assume full state is available for measurement but it needs an observer. Of course, it does not assume exact knowledge of either robot or actuator parameters. The optimality is on

the basis of the minimization of a performance index which turns out to be possible if we could find a solution to the Hamilton Jacobi equation.

(Aiko and Kimura, 2002) aim to establish control theoretical validity of the feedback error learning scheme suggested as an architecture of brain motor control with deep physiological root in computational neuroscience. The feedback error learning method is formulated as a two-degree of freedom adaptive control. The stability of the adaptive control law is shown clearly based on the strict positive realness, under the supposition that the plant is stable and stably invertible. Results prove the effectiveness of the method.

(McLain and Henson, 2000) present a nonlinear adaptive control strategy established upon radial basis function networks and principal component analysis. The suggested method is well suited for low dimensional nonlinear systems that are difficult to model and control via conventional means. The effective system dimension is decreased by applying nonlinear principal component analysis to state variable data obtained from open-loop tests. This permits the radial basis functions to be placed in a lower dimensional space than the original state space. The total number of basis functions is clearly described a priori, and an algorithm which adjusts the place of the basis function centers to encompass the current operating point is presented. The basis function weights are adapted on-line such that the plant output asymptotically follows a linear reference model. A highly nonlinear polymerization reactor is used to compare the nonlinear adaptive controller to a linear state feedback controller that takes advantage of the same amount of plant information.

(Makoudi and Radouane 2000) present a distributed model reference adaptive control for interconnected subsystems in the sense that no information exchange occurs between the subsystems. The approach is established upon the interconnection output estimation using the polynomial series which suggests a general solution for interconnected subsystems. The parameter estimation scheme is an integrated adaptive data filtering with a recursive least-squares algorithm with parameter projection and normalization. The problem of minimum phased subsystems is handled by an adaptive input output data filtering. Hence the zeros of each subsystem estimated model are replaced inside the unit circle. This estimated model which is minimum phased is then used for the control synthesis. It is shown that the stability conditions established upon weak interconnections are relaxed. Also the robustness of the

suggested adaptive control against unmodeled dynamics is expressed. At last, the results are illustrated by numerical examples.

(Tsai and Lin, 1997) present a model reference adaptive control approximation, it is formed in the modal space; it is applied for flutter control of a cantilever pipe conveying fluid. The control input is supplied by a pair of surface mounted piezoelectric actuators which are driven 180 ^{o} out of phase to provide an equivalent bending moment acting on the controlled system. Comparison of performance of the model reference adaptive control with that of the optimal independent modal space control shows that the former is more robust than the latter in terms of flow speed variations, which are unknown in the control system designed ; that is, the adaptive approach can compensate a larger range of flow speed uncertainties without resulting in an unstable control system, hence successful flutter suppression of the fluid conveying cantilever pipe with high flow speed can be performed.

(Lee at all, 1998) present that an adaptive neural network full state feedback controller has been designed and applied to the passive line of sight (LOS) stabilization system. Model reference adaptive control (MRAC) is well founded for linear systems. However, this method cannot be utilized directly because the LOS system is nonlinear in nature. Utilizing the universal approximation property of neural networks, an adaptive neural network controller is presented by generalizing the model reference adaptive control technique, in which the gains of the controller are approached by neural networks. This ejects the requirement of linearizing the dynamics of the system, and the stability properties of the closed loop system can be satisfied.

(Taware at all, 2003) present that friction correction for a benchmark system with load friction plus joint flexibility and damping is addressed. This is a difficulty of controlling a sandwich dynamic system with a non-smooth nonlinearity. Few non-adaptive and adaptive compensation designs are analyzed, established upon a state feedback output tracking model reference adaptive control scheme. Adequate output matching conditions are derived for friction compensation. Approximate linear parameterizations of nonlinear friction are built for adaptive friction compensator designs. Simulation results confirm the desired system performance.

(Zhong, 2005) presents that model reference adaptive control problem for single input single output time invariant continuous time plants with input saturation is taking into account

with main attention focused on global properties. A sufficient condition is presented and a new design method of adaptive control systems is suggested. If a priori knowledge about the plant is available to choose the reference model and the reference input so that the sufficient condition holds, the closed loop adaptive control system designed by the suggested method can have global stability and globally output tracking property. It is shown that the sufficient condition is necessary in some cases.

(Chien and Yao , 2004) present that a model reference adaptive control strategy is used to design an iterative learning controller for a class of repeatable nonlinear systems with uncertain parameters, high relative degree, initial output resetting error, input disturbance and output noise. The class of nonlinear systems should gratify some differential geometric conditions such that the plant can be transformed via a state transformation into an output feedback canonical form. An appropriate error model is derived based on signals filtered from plant input and output. The learning controller compensates for the unknown parameters, uncertainties and nonlinearity by means of projection type adaptation laws which update control parameters along the iteration domain. It is shown that the internal signals stay bounded for all iterations. The output tracking error will converge to a profile which can be adjusted by design parameters and the learning speed is increased if the learning gain is large.

(Costa at all, 2003) present that the design of Model Reference Adaptive Control for Multi Input Multi Output (MIMO) linear systems has not yet achieved, despite significant efforts, the completeness and simplicity of its Single Input Single Output (SISO) counterpart. One of the main obstructions has been the generalization of the SISO assumption that the sign of the high frequency gain (HFG) is known. Here they overcome this obstacle and present a more complete MIMO analog to the renowned Lyapunov based SISO design which is significantly less restrictive than the existing analogs. Their algorithm makes use of a new control parameterization derived from a factorization of the HFG matrix $K_p = SDU$, where S is positive symmetric definite, D is diagonal, and U is unity upper triangular. Only the signs of the entries of D or, equally, the signs of the leading principal minors of K_p, are assumed to be known.

(Krstic and Banaszuk, 2005) consider a class of Multi Input Multi Output (MIMO) LTI models with uncertain resonant modes and time delays, which are common in control of instabilities arising in jet engines. With uncertain delays preventing the use of model reference adaptive control, they develop an adaptive MIMO pole placement scheme for the

system. They use indirect adaptation, estimating a small number of physical parameters from a nonlinearly parameterized plant. To address the highly noisy environment in jet engines they introduce the dead zone in the adaptation law and present simulations that successfully stabilize the system in the presence of noise and serious actuator saturation.

(WANG at all, 1997) present novel approach for the fault detection and diagnosis (FDD) of faults in actuators and sensors by way of the use of adaptive updating rules. The system considered is linear time invariant and is subjected to an unknown input that represents either model uncertainty or immeasurable disturbances. Firstly, fault detection and diagnosis for linear actuators and sensors are considered, where a fixed observer is used to detect the fault while an adaptive diagnostic observer is built to diagnose the fault. Utilizing the augmented error technique from mode1 reference adaptive control, an observation error model is formulated and used to establish an adaptive diagnostic algorithm that produces an estimate of the gains of actuator and the sensor. An extension to the fault detection and diagnosis to include nonlinear actuators is also made, where a similar augmented error model to that used for linear actuators and sensors is acquired. As a result, a convergent adaptive diagnostic algorithm for estimating the parameters in the nonlinear actuators is improved.

(Sinha and Pechev, 1999) present a model reference adaptive controller (MRAC) for magnetically suspended vehicles (maglev) using the criterion of stable maximum descent. The adaptation algorithm is forced to reduce the air gap error between the reference model and the actual system. The explicit relationship between the parameters of the performance criterion (function of the air gap error and its derivative) and the state feedback adaptation rule is produced for a single degree of freedom suspension system. Experimental results from a small representative test rig are presented to illustrate the efficiency of the suggested non linear controller in the presence of variations in pay load (suspended mass), disturbance force and air gap set point. Hardware aspects of the transporter and Digital Signal Processing (DSP) based real time controller are briefly discussed to emphasize some of the practical issues related to digital execution of the air gap adaptive control law.

(Makoudi and Radouane, 1999) present a decentralized model reference adaptive control (DMRAC) for interconnected subsystems with unknown or time-varying time delay. The decentralization approximation is established upon the interconnection output estimation using the polynomial series which suggests a general solution for interconnected subsystems. The parameter estimation scheme is a combined adaptive data filtering with a recursive least squares algorithm with parameter projection and signal normalization. A “good data” model

is determined by an adaptive filtering of the input and output signals. The acquired model allows dealing with non-minimum phased subsystems with unknown or time-varying dead time and at the same time to relax the hypothesis of weak interconnections for decentralized control.

(Tian and Hoo, 2003) present that transition control is determined as a type of control method that is operated when the plant transitions from one steady state to another as a result of a set point change. Recent approaches have depended on multiple models and centralized or decentralized controller designs to address this issue. This work presents and improves a transition control framework that consists of multiple fixed and adaptive models within a state-shared non minimal realization and an H_∞ controller design. The effectiveness of this transition control framework is presented on two nonlinear single-input single-output reactors in the face of modeling errors, parameter uncertainties and disturbances.

(Chaoa and Neoub, 2000) present that model reference adaptive control of air-lubricated capstan drive for precision positioning. Because friction-induced nonlinearities in positioning systems are mostly range of motion-dependent, dual-model or dual-stage strategies are frequently adopted to deal with the incompatibility encountered when a system moves from submicrometer steps (micro mode) to larger scale strokes (macro mode). Despite the fact that good performance is usually acquired when each stage functions in its designed range of motion, a system often performs less sufficiently when operating near the switching point between models or stages. An air-lubricated capstan drive was used in this work to minimize the inconsistency between macro and micro modes, and a single mode MRAC was designed to control the capstan drive system for precision positioning. Accuracy better than 615 nm with no overshooting was obtained in all conditions tested (including 50 nm, 500 nm and 10 mm steps).

(Mirkin and Gutman, 2005) present that two new output feedback adaptive control schemes established upon Model Reference Adaptive Control (MRAC) and adaptive laws for updating the controller parameters are improved for a class of linear MIMO systems with state delay. An effective controller structure established upon a new error equation parameterization is suggested to achieve tracking with the error tending to zero asymptotically. To accomplish exact asymptotical tracking, they introduce, in the standard MRAC structure for plants without delay, a new supplemental adaptive feedforward control component as an output of a dynamical system driven by the reference signal. Adaptive laws are improved using the SPR-Lyapunov design approximation and two suppositions regarding the previous knowledge of the high frequency matrix K_p. This study is the first asymptotic

exact zero tracking results for this class of systems in the framework of the certainty equivalence approximation.

In this thesis the standard model reference adaptive control is used to control the position of a permanent magnet dc motor, some of its parameters can not be measured directly. Further, some of the other parameters are not constant and vary as the motor operates. A second order system is used as the reference model.

This thesis is organized as follows: Fundamentals of MRAC is summarized in Chapter 2. Model reference adaptive control for systems without finite transmission zeros is introduced in Chapter 3. MRAC for dc motor is given in Chapter 4. Simulation results are presented in Chapter 5. Finally conclusions are made in Chapter 6.

CHAPTER II

MODEL REFERENCE ADAPTIVE CONTROL

2.1 GENERAL CONTROL SYSTEM DESIGN STEPS

The steps in a general control design problem are shown in Figure 2.1. Each step is explained below (Ioannou, 1994).

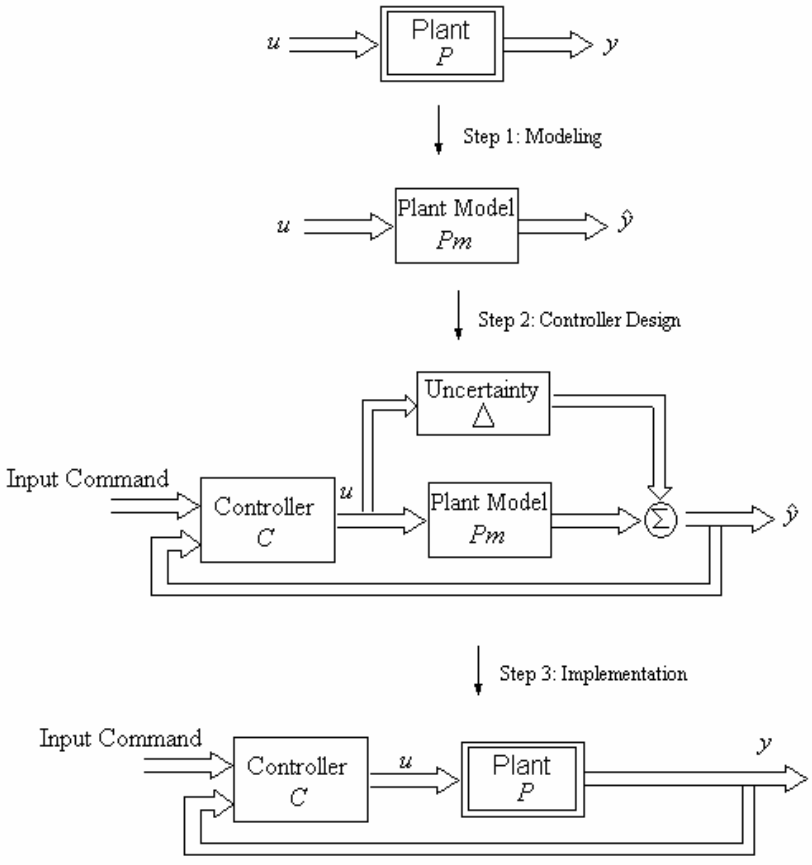

Figure 2.1 General control system design steps.

Step 1. Modeling

A plant model may be built by using physical laws or by processing the plant input output (I/O) data obtained by performing various experiments. But, this model may be complex for the controller design and additional simplifications may be necessary. Some of the approaches in many cases used to obtain a simplified model are

(i) Linearization around functioning points,

(ii) Model order decreasing techniques.

In approximation (i) the plant is approximated by a linear model that is acceptable around a given operating point. Different functioning points may lead to some different linear models that are used as plant models. Linearization is performed by using Taylor's series expansion and approximation, fitting of empirical data to a linear model, etc.

In approach (ii) small impacts and phenomena outside the frequency range of interest are disregarded leading to a lower order and uncomplicated plant model.

Step 2. Controller Design

The controller is planned to meet the performance necessities for the plant model. Δ symbolizes most of the unmodeled plant phenomena. The control engineer may be able to modify or redesign the controller to be more robust with respect to Δ. This robustness analysis and redesign increases the ability for a successful execution in Step 3.

Step 3. Implementation

The implementation can be done using a digital computer. The type of computer, the type of interface devices between the computer and the plant, software tools are considered priority matters. Computer speed and preciseness limitations may constrain on the complexity of the controller. It may force the control engineer to go back to Step 2 or even Step 1. Other important aspect of implementation is the final adjustment, or as often called the tuning which is often done by trial and error and depends very much on the experience and intuition of the control engineer.

2.2 ADAPTIVE CONTROL

The words “adaptive systems" and “adaptive control" have been employed as early as 1950 (Ioannou, 1994). The design of autopilots for high-performance airplane was one of the first incentives for active investigation on adaptive control in the early 1950s. Airplane function over a broad range of speeds and heights, and its dynamics are nonlinear and conceptually time varying. For a given functioning point, described by the airplane speed and height, the complex airplane dynamics can be approached by a linear model of the similar shape as (2.1).

$$\dot{x} = Ax + Bu\,; \quad x(0) = x_0\,, y = C^T x + Du \quad . \tag{2.1}$$

For instance, for a functioning point i, the linear airplane model has the following shape:

$$\dot{x} = A_i x + B_i u\,; \quad x(0) = x_0\,, y = C_i^{\,T} x + D_i u \tag{2.2}$$

where x, u, y are the state, input, output vectors, respectively; and $A_i\,, B_i\,, C_i$ and D_i are coefficient matrix functions of the functioning point i. As the airplane moves through different flight situations, the functioning point changes goes to different values for $A_i\,, B_i\,, C_i$ and D_i. Since the output response y (t) transports information about the state x as well as the parameters, one may dispute that in principle, a complicated feedback controller should be able to learn about parameter alterations by processing y (t) and utilize the suitable gains to adapt them. This quarrel goes to a feedback control structure on which adaptive control is established. The controller building comprise of a feedback loop and a controller with adjustable gains as shown in Figure 2.2.

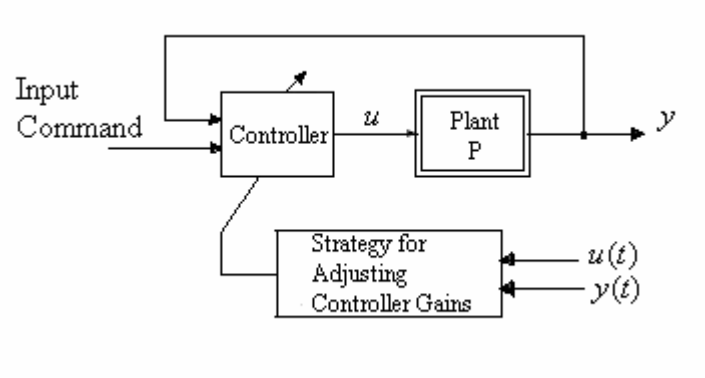

Figure 2.2 Controller building with adjustable controller gains.

2.2.1 Direct and Indirect Adaptive Control

An adaptive controller is shaped by combining an on-line parameter estimator, which gives estimates of unknown parameters at each moment, with a control law that is prompted from the known parameter case. The method of the parameter estimator, also referred to as adaptive law, is joined with the control law lead to two different approximations. In the first approximation, mentioned as indirect adaptive control, the plant parameters are estimated on-line and used to compute the controller parameters. This approximation has also been mentioned as explicit adaptive control, due to the fact that the design is on the foundation of an explicit plant model.

In the second approximation, mentioned as direct adaptive control, the plant model is parameterized regarding the controller parameters that are estimated directly without intermediate computations involving plant parameter estimates. This approximation has also been mentioned as implicit adaptive control due to the fact that the design is established upon the estimation of an implicit plant model.

In indirect adaptive control, the plant model $\mathrm{P}(\theta^*)$ is parameterized with regards to some unknown parameter vector θ^*. For instance, for a linear time invariant (LTI) SISO plant model, θ^* may symbolize the unknown coefficients of the numerator and denominator of the plant model transfer function. An on-line parameter estimator generates an estimate $\theta(\mathrm{t})$ of θ^* at each time t by working the plant input u and output y. The parameter estimate $\theta(\mathrm{t})$ clearly describes an estimated plant model characterized by $\hat{\mathrm{P}}(\theta(\mathrm{t}))$ that for control design aims is treated as the “true" plant model and is used to calculate the controller parameter $\theta_C(\mathrm{t})$ by solving a specific algebraic equation $\theta_C(\mathrm{t}) = \mathrm{F}(\theta(\mathrm{t}))$ at every time t. The

shape of the control law $C(\theta_C)$ and algebraic equation $\theta_C = F(\theta)$ is selected to be identical to that of the control law $C(\theta_C^*)$ and equation $\theta_C^* = F(\theta^*)$ that could be used to encounter the performance necessities for the plant model $P(\theta^*)$ if θ^* was known. It is, hence, clear that with this approximation, $C(\theta_C(t))$ is designed at every time t to gratify the performance necessities for the estimated plant model $\hat{P}(\theta(t))$, which may not be the same from the unknown plant model $P(\theta^*)$. And so, the head problem in indirect adaptive control is to select the class of control laws $C(\theta_C)$ and the class of parameter estimators that produce $\theta(t)$ and likewise the algebraic equation $\theta_C(t) = F(\theta(t))$ so that $C(\theta_C(t))$ encounters the performance necessities for the plant model $P(\theta^*)$ with unknown θ^*. The block diagram of an indirect adaptive control scheme is seen in Figure 2.3.

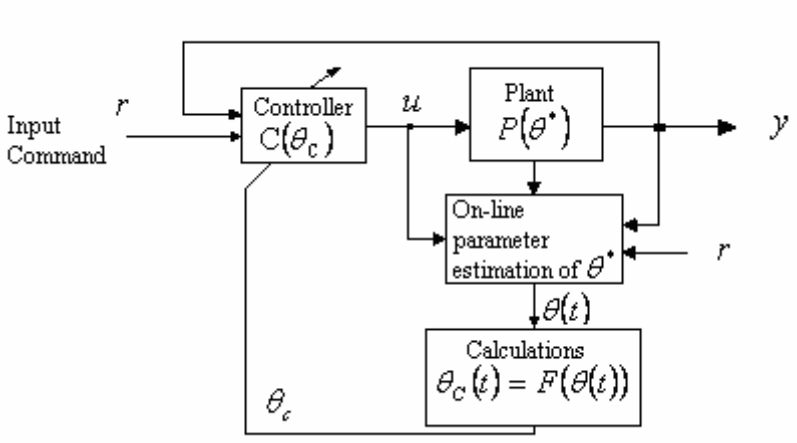

Figure 2.3 Indirect adaptive control.

In direct adaptive control, the plant model $P(\theta^*)$ is parameterized in the sense of the unknown controller parameter vector θ_C^*, for which $C(\theta_C^*)$ encounters the performance necessities, to acquire the plant model $P_C(\theta_C^*)$ with precisely the similar input output characteristics as $P(\theta^*)$. The on-line parameter estimator is designed based on $P_C(\theta_C^*)$ in place of $P(\theta^*)$ to supply direct estimates $\theta_C(t)$ of θ_C^* at every time t by working the plant input u and output y. The estimate $\theta_C(t)$ is then used to update the controller parameter vector θ_C without intermediate computations. The selection of the class of control laws $C(\theta_C)$ and parameter estimators producing $\theta_C(t)$ for which $C(\theta_C(t))$ encounters the

performance necessities for the plant model P(θ^*) is the basic problem in direct adaptive control. The characteristics of the plant model P(θ^*) are very important in acquiring the parameterized plant model $P_C(\theta_C^*)$ that is useful for on-line estimation. Due to that, direct adaptive control is constrained to a specific class of plant models. A class of plant models that is appropriate for direct adaptive control comprise of all SISO plant models that are minimum-phase, i.e., their zeros are placed in Re [s] < 0. The block diagram of direct adaptive control is seen in Figure 2.4.

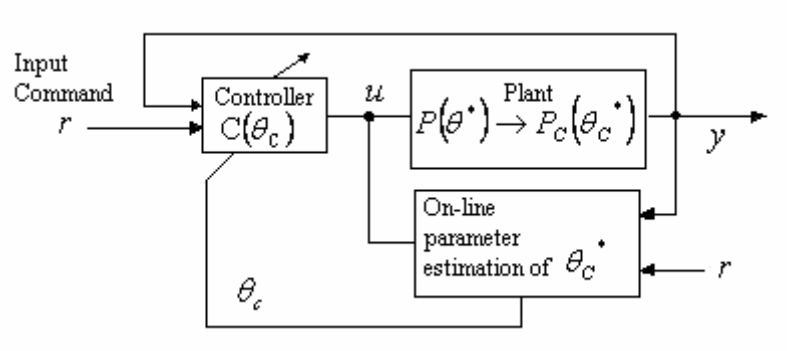

Figure 2.4 Direct adaptive control**.**

The foundation behind the design of direct and indirect adaptive control seen in Figures 2.3 and 2.4 is conceptually simple. The design of $C(\theta_C)$ acts the estimates $\theta_C(t)$ (regarding direct adaptive control) or the estimates $\theta(t)$ (regarding adaptive control) as if they were the true parameters. This design approximation is named certainty equivalence and can be used to produce a broad class of adaptive control schemes by joining different on-line parameter estimators with different control laws.

The idea in back of the certainty equivalence approximation is that as the parameter estimates, $\theta_C(t)$ and $\theta(t)$ converge to the true ones θ_C^* and θ^*, respectively, the performance of the adaptive controller $C(\theta_C)$ be inclined to that accomplished by $C(\theta_C^*)$ concerning known parameters.

2.2.2 Model Reference Adaptive Control

Model reference adaptive control (MRAC) comes from the model following problem or model reference control (MRC) problem. In MRC, a good comprehension of the plant and the performance necessities it has to encounter permit the designer to invent a model, mentioned as the reference model, that depicts the desired I/O characteristics of the closed-loop plant. The purpose of MRC is to discover the feedback control law that alters the structure and dynamics of the plant so that its I/O characteristics are precisely the same as those of the reference model. The building of an MRC diagram for a LTI, SISO plant is shown in Figure 2.5. The transfer function $W_M(s)$ of the reference model is designed so that for a given reference input signal $r(t)$ the output $y_m(t)$ of the reference model depicts the wanted response the plant output $y(t)$ must follow.

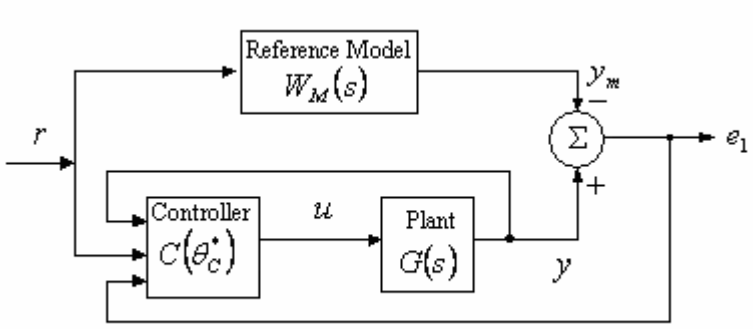

Figure 2.5 Model reference control.

The feedback controller symbolized by $C(\theta_C^*)$ is designed so that all signals are delimited and the closed-loop plant transfer function from r to y is equal to $W_M(s)$. This transfer function matching assures that for any given reference input $r(t)$, the tracking error $e_1 \cong y - y_m$, which symbolizes the divergence of the plant output from the desired trajectory y_m, approaches to zero with time. The transfer function matching is accomplished by canceling the poles of the plant transfer function $G(s)$ and substituting them with those of $W_M(s)$ through the use of the feedback controller $C(\theta_C^*)$. The cancellation of the plant poles brings a limitation on the plant to be minimum phase, in other words, have stable poles. When the plant poles are not stable, its cancellation may easily go to uncontrolled signals.

The design of $C(\theta_C^*)$ necessitates the learning of the coefficients of the plant transfer function $G(s)$. If θ^* is a vector including all the coefficients of $G(s) = G(s;\theta^*)$, then the parameter vector θ_C^* may be calculated by solving an algebraic equation of the shape

$$\theta_C^* = \mathrm{F}(\theta^*) \quad . \tag{2.3}$$

Finally for the realization of MRC, the plant model has to be minimum phase and its parameter vector θ^* has to be known precisely.

When θ^* is not known the MRC scheme of Figure 2.5 can not be executed because θ_C^* can not be computed applying Eq. 2.3. One method of referring to the unknown parameter matter is to utilize the certainty equivalence approximation to substitute the unknown θ_C^* in the control law with its estimate $\theta_C(t)$ acquired using the direct or the indirect approach. The deriving control schemes are accepted as MRAC and can be classified as indirect MRAC seen in Figure 2.6 and direct MRAC seen in Figure 2.7.

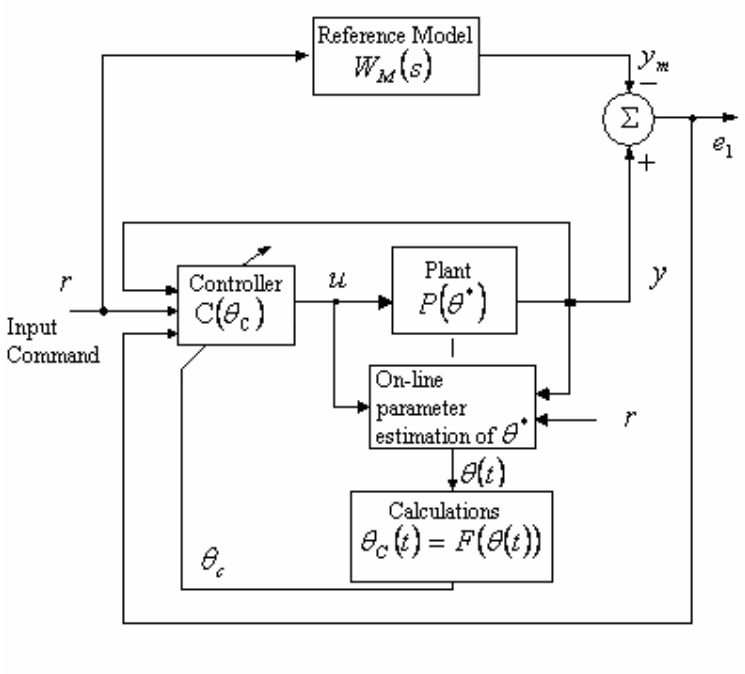

Figure 2.6 Indirect MRAC.

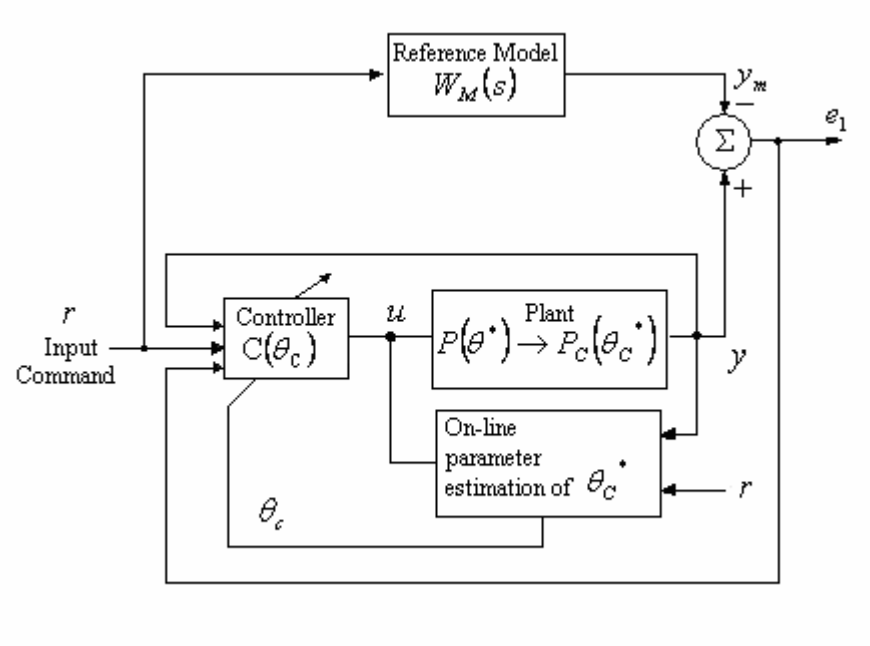

Figure 2.7 Direct MRAC.

In this thesis, direct MRAC approach is used for the control of a permanent magnet dc motor which does not have finite transmission zeros and the method for this is discussed in the following chapter.

CHAPTER III

MRAC FOR SYSTEMS WITHOUT FINITE TRANSMISSION ZEROS

The aim of this method realizes model reference adaptive control for the plant that has single input single output and without finite transmission zeros. Adaptation mechanism adjusts feedback gains of states and the feed-forward gain of reference signal. The coefficients which are in the transfer function of closed loop plant are equalized to model's.

3.1 THE STRUCTURE OF THE ADAPTIVE CONTROL SYSTEM

The structure of the adaptive control system is given Figure 3.1 (Karadeniz et al, 2004). In this figure r is reference signal , u is plant's input , g is feed forward gain , $\mathbf{F}=[F_1\ F_2.....F_n]^T$ is feedback gains of plant's states, $\boldsymbol{x}$ and $\boldsymbol{z}$ are controllable canonical form of plant and model's states respectively, $\boldsymbol{e}$ is error signal between model and plant states. Plant follow model via $\boldsymbol{F}(t)$ and $g(t)$ which are updated every time step.

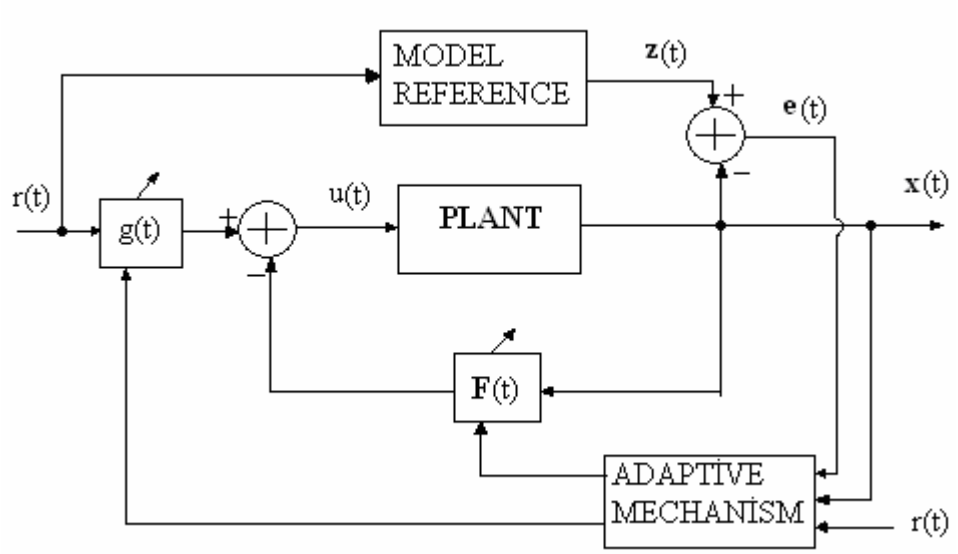

Figure 3.1 The structure of the adaptive control system.

The transfer function of one input one output continuous time plant without finite transmission zeros is

$$\frac{K}{s^n + a_n s^{n-1} + ... + a_2 s + a_1} \quad . \tag{3.1}$$

The controllable canonical form of (3.1) is

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ ..... \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -a_1 & -a_2 & -a_3 & \dots & -a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ ..... \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dots \\ K \end{bmatrix} u(t) \ , \tag{3.2 -a}$$

$$y = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \boldsymbol{x} \ , \tag{3.2 -b}$$

which is in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u \ , \tag{3.3 -a}$$

$$y = x_1 \ . \tag{3.3 -b}$$

As shown in this figure, the following feedback control law used is

$$u = gr - \boldsymbol{F}^T \boldsymbol{x} \ . \tag{3.4}$$

We can show the controlled plant's state space equation is

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{b}\left(g\,r - \boldsymbol{F}^T \boldsymbol{x}\right) = \left(\boldsymbol{A} - \boldsymbol{b}\boldsymbol{F}^T\right)\boldsymbol{x} + \boldsymbol{b}gr \ , \tag{3.5-a}$$

Or most explicitly,

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ ..... \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -a_1 - KF_1 & -a_2 - KF_1 & -a_3 - KF_1 & \dots & -a_n - KF_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ ..... \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dots \\ Kg \end{bmatrix} r(t) \ . \tag{3.5-b}$$

The reference model transfer function and state space equation are

$$\frac{K^*}{s^n + a_n^* s^{n-1} + ... + a_2^* s + a_1^*} \quad , \tag{3.6-a}$$

$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \\ \dots\dots \\ z_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -a_1^* & -a_2^* & -a_3^* & \dots & -a_n^* \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \dots\dots \\ z_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \dots \\ K^* \end{bmatrix} u(t) \qquad (3.6\text{-b})$$

which is in the form

$$\dot{\boldsymbol{z}} = \boldsymbol{A_m z} + \boldsymbol{b_m}\, r\,. \qquad (3.6\text{-c})$$

Assume the model output

$$y_m = z_1\,. \qquad (3.6\text{-d})$$

3.2 DERIVATION OF THE ADAPTIVE LAWS

Similar adaptive laws used in (Karadeniz et al, 2004) are followed in the sequel. From Eqs. 3.5-a and 3.6-c, for the equivalence of the controlled plant and the reference model

$$\boldsymbol{A_m} = \boldsymbol{A} - \boldsymbol{b}\,\boldsymbol{F}^{*T} \quad , \qquad (3.7\text{-a})$$

$$\boldsymbol{b_m} = \boldsymbol{b}\, g^* \,, \qquad (3.7\text{-b})$$

or explicitly

$$\boldsymbol{F}_j = \boldsymbol{F}_j^* = \frac{a_j^* - a_j}{K} \,, \qquad (j = 1,2,....,n) \,, \qquad (3.8\text{-a})$$

$$g = g^* = \frac{K^*}{K} \,. \qquad (3.8\text{-b})$$

Normally, a_j and K are not known. Resultantly, F_j^* and $g*$ are not known. Therefore F_j and g may be different from F_j^* and $g*$. The purpose of the adaptive law is to converge the present values of F_j and g to F_j^* and g*, respectively, i.e,

$$\lim_{t \longrightarrow \infty} \boldsymbol{F}(t) \longrightarrow \boldsymbol{F}^*, \qquad (3.9\text{-a})$$

$$\lim_{t \longrightarrow \infty} g(t) \longrightarrow g^*. \qquad (3.9\text{-b})$$

Define the error by

$$\boldsymbol{e} = \boldsymbol{z} - \boldsymbol{x}\,. \qquad (3.10\text{-a})$$

Then, derivative of the error is

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{z}} - \dot{\boldsymbol{x}} \ . \tag{3.10-b}$$

We manipulate the plant and model equations in (3.4), (3.5) and (3.6), respectively and obtain

$$\begin{aligned}
\dot{\boldsymbol{e}} &= \left(\boldsymbol{A}_m\, \boldsymbol{z} + \boldsymbol{b}_m\, r\right) - \left(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{b}\,u\right) \\
&= \left(\boldsymbol{A}_m\, \boldsymbol{z} + \boldsymbol{b}_m\, r\right) - \left(\boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{b}\left(rg - \boldsymbol{F}^T \boldsymbol{x}\right)\right) \\
&= \boldsymbol{A}_m\, \boldsymbol{z} + \boldsymbol{b}_m\, r - \boldsymbol{A}\,\boldsymbol{x} - \boldsymbol{b}rg + \boldsymbol{b}\boldsymbol{F}^T \boldsymbol{x} \\
&= \boldsymbol{A}_m\, \boldsymbol{z} + \boldsymbol{b}_m\, r - \left(\boldsymbol{A}_m + \boldsymbol{b}\boldsymbol{F}^{*T}\right)\boldsymbol{x} - \boldsymbol{b}rg + \boldsymbol{b}\boldsymbol{F}^T \boldsymbol{x} \\
&= \boldsymbol{A}_m \left(\boldsymbol{z} - \boldsymbol{x}\right) + \boldsymbol{b}_m\, r + \boldsymbol{b}\left(\boldsymbol{F}^T - \boldsymbol{F}^{*T}\right)\boldsymbol{x} - \boldsymbol{b}rg \\
&= \boldsymbol{A}_m \boldsymbol{e} + \frac{1}{\mathrm{g}^*} \boldsymbol{b}_m \left(\boldsymbol{F}^T - \boldsymbol{F}^{*T}\right)\boldsymbol{x} + \boldsymbol{b}_m\, r - \frac{\boldsymbol{b}_m rg}{\mathrm{g}^*} \\
&= \boldsymbol{A}_m \boldsymbol{e} + \frac{1}{g^*} \boldsymbol{b}_m \left(\boldsymbol{F}^T - \boldsymbol{F}^{*T}\right)\boldsymbol{x} + \boldsymbol{b}_m\ r \frac{1}{g^*}\left(g^* - g\right).
\end{aligned} \tag{3.11}$$

Define:

$$\boldsymbol{\Phi}_{\boldsymbol{F}} = \boldsymbol{F} - \boldsymbol{F}^* \ , \tag{3.12-a}$$

$$\Phi_g = g^* - g \ . \tag{3.12-b}$$

Then,

$$\dot{\boldsymbol{\Phi}}_F = \dot{\boldsymbol{F}} \ , \tag{3.12-c}$$

$$\dot{\Phi}_g = -\dot{g} \ . \tag{3.12-d}$$

We renew the derivative of the error equation in (3.11),

$$\begin{aligned}
\dot{\boldsymbol{e}} &= \boldsymbol{A}_m \boldsymbol{e} + \frac{1}{g^*} \boldsymbol{b}_m \boldsymbol{\Phi}_{\boldsymbol{F}}^{\boldsymbol{T}} \boldsymbol{x} + \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m \ , \\
&= \boldsymbol{A}_m \boldsymbol{e} + \frac{1}{g^*} \boldsymbol{b}_m \boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{\Phi}_{\boldsymbol{F}} + \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m \ .
\end{aligned} \tag{3.13}$$

We suggest Lyapunov's method [1] for updating of **F** and g. For this we define the Lyapunov function

$$\boldsymbol{V} = \boldsymbol{e}^T P \boldsymbol{e} + \frac{1}{\alpha\, g^*} \boldsymbol{\Phi}_F^T \boldsymbol{\Phi}_F + \frac{1}{\alpha\, g^*} \Phi_g^2 \ , \tag{3.14}$$

where $\boldsymbol{P}$ is strictly positive real matrix to be found, α is a positive real coefficient.

[1] Extra knowledge with relating to Lyapunov Theorem is placed in Appendix A.

Assuming g* > 0, Lyapunov function satisfies $V \geq 0$.

Derivative of the Lyapunov's function can be found from Eq. 3.14 and using Eq. 3.13

$$\dot{\boldsymbol{V}} = \frac{d}{dt}\left(\boldsymbol{e}^T \boldsymbol{P}\, \boldsymbol{e}\right) + \frac{1}{\alpha\, g^*}\boldsymbol{\Phi}_F^T \dot{\boldsymbol{\Phi}}_F + \frac{1}{\alpha\, g^*}\dot{\boldsymbol{\Phi}}_F^T \boldsymbol{\Phi}_F + \frac{2}{\alpha\, g^*}\dot{\boldsymbol{\Phi}}_g \boldsymbol{\Phi}_g \ . \qquad (3.15\text{-a})$$

Where, first term of Eq.3.15-a can be rewritten as shown in Eq.3.15-b

$$\begin{aligned}
&\frac{d}{dt}\left(\boldsymbol{e}^T \boldsymbol{P} \boldsymbol{e}\right) = \dot{\boldsymbol{e}}^T \boldsymbol{P} \boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P} \dot{\boldsymbol{e}} \\
&= \left(\boldsymbol{A}_m \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{b}_m \boldsymbol{x}^T \boldsymbol{\Phi}_F + \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m\right)^T \boldsymbol{P} \boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\left(\boldsymbol{A}_m \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{b}_m \boldsymbol{x}^T \boldsymbol{\Phi}_F + \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m\right) \\
&= \left(\boldsymbol{e}^T \boldsymbol{A}_m^T + \frac{1}{g^*}\boldsymbol{\Phi}_F^T \boldsymbol{x} \boldsymbol{b}_m^T + \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m^T\right) \boldsymbol{P} \boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P}\left(\boldsymbol{A}_m \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{b}_m \boldsymbol{x}^T \boldsymbol{\Phi}_F + \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m\right) \\
&= \boldsymbol{e}^T \boldsymbol{A}_m^T \boldsymbol{P} \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{\Phi}_F^T \boldsymbol{x} \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e} + \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e} + \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{A}_m \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b}_m \boldsymbol{x}^T \boldsymbol{\Phi}_F + \frac{1}{g^*} r \Phi_g \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b}_m \\
&= \boldsymbol{e}^T \left(\boldsymbol{A}_m^T \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A}_m\right) \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{\Phi}_F^T \boldsymbol{x} \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b}_m \boldsymbol{x}^T \boldsymbol{\Phi}_F + \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e} + \frac{1}{g^*} r \Phi_g \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b}_m
\end{aligned}$$

3.15 – b

Eq. 3.15-b is replaced in to Eq. 3.15-a. So that, the derivative of Lyapunov's function is acquired as

$$\begin{aligned}
\dot{\boldsymbol{V}} &= \boldsymbol{e}^T \left(\boldsymbol{A}_m^T \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A}_m\right) \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{\Phi}_F^T \boldsymbol{x} \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e} + \frac{1}{g^*}\boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b}_m \boldsymbol{x}^T \boldsymbol{\Phi}_F \\
&+ \frac{1}{g^*} r \Phi_g \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e} + \frac{1}{g^*} r \Phi_g \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b}_m + \frac{1}{\alpha g^*}\boldsymbol{\Phi}_F^T \dot{\boldsymbol{\Phi}}_F + \frac{1}{\alpha g^*}\dot{\boldsymbol{\Phi}}_F^T \boldsymbol{\Phi}_F + \frac{2}{\alpha g^*}\dot{\Phi}_g \Phi_g . \qquad (3.15\text{–c})
\end{aligned}$$

The matrix of the first quadratic form in Eq. 3.15-c is chosen as

$$\boldsymbol{A}_m^T \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A}_m = -\boldsymbol{Q} \qquad (3.16)$$

which is known as Lyapunov Equation where, the eigenvalues of $\boldsymbol{A}_m$ are negative for a stable reference model and the matrix Q is chosen as positive definite symmetric matrix. So, the solution of Lyapunov Equation gives us positive definite symmetric matrix $\boldsymbol{P}$. Thus the first term of Eq. 3.15-c $\boldsymbol{e}^T \left(\boldsymbol{A}_m^T \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A}_m\right) \boldsymbol{e}$ is made negative definite. If, the remaining terms of Eq.

3.15-c cancel each other, $\dot{\boldsymbol{V}} < 0$ and stability is acquired. Therefore, the following equations must be realized for cancellation :

$$\dot{\boldsymbol{\Phi}}_F^T \boldsymbol{\Phi}_F = -\alpha \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b}_m \boldsymbol{x}^T \boldsymbol{\Phi}_F, \tag{3.17-a}$$

$$\boldsymbol{\Phi}_F^T \dot{\boldsymbol{\Phi}}_F = -\alpha \boldsymbol{\Phi}_F^T \boldsymbol{x} \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e} \ , \tag{3.17-b}$$

$$2\dot{\Phi}_g \Phi_g = -\alpha r \Phi_g \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e} - \alpha r \Phi_g \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{b}_m \ . \tag{3.17-c}$$

Eqs. (3.17-a) and (3.17-b) are equivalent. To satisfy these equations, the parameter $\dot{\boldsymbol{\Phi}}_F$ can be chosen

$$\dot{\Phi}_F = -\alpha \boldsymbol{x}(t) \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e}(t). \tag{3.18-a}$$ To satisfy Eq. (3.17-c), the parameter $\dot{\Phi}_g$ can be set

$$\dot{\Phi}_g = -\alpha r(t) \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e}(t). \tag{3.18-b}$$

The adaptive laws to update $\boldsymbol{F}$ and g are finally obtained from Eqs. 3.18-a, b together with Eqs. 3.12 c, d as follows

$$\dot{\boldsymbol{F}} = -\alpha \boldsymbol{x}(t) \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e}(t) \ , \tag{3.18-c}$$

$$\dot{g} = \alpha r(t) \boldsymbol{b}_m^T \boldsymbol{P} \boldsymbol{e}(t) \ . \tag{3.18-d}$$

Hence Eqs 3.18-c and 3.18-d are the adaptive laws which are chosen. So, with these choices, the derivative of Lyapunov function in Eq. 3.15 c is negative definite, that is

$$\dot{\boldsymbol{V}} = -\boldsymbol{e}^T \boldsymbol{Q} \boldsymbol{e} < 0 \tag{3.19}$$

for the nonzero error vector ***e.***

This implies

$$\lim_{t \to \infty} V(t) = 0 \ . \tag{3.19-a}$$

From Eq. 3.14, Eq. 3.19-a implies

$$\lim_{t \to \infty} \boldsymbol{\Phi}_F = 0 \ , \tag{3.19-b}$$

$$\lim_{t \to \infty} \Phi_g = 0 \ . \tag{3.19-c}$$

Consequently, Eq. 3.12 a and b imply

$$\lim_{t \to \infty} \boldsymbol{F} \to \boldsymbol{F}^* , \tag{3.19-d}$$

$$\lim_{t \to \infty} g \to g^* , \tag{3.19-e}$$

respectively. With these equations, Eq. 3.11 becomes

$$\dot{\boldsymbol{e}} = \boldsymbol{A}_m \boldsymbol{e} \tag{3.19-f}$$

Therefore,

$$\lim_{t\to\infty} \boldsymbol{e} \to 0 \ , \tag{3.19-h}$$

since $\boldsymbol{A}_m$ is a stable matrix. Finally, Eq. 3.19-h leads to

$$\lim_{t\to\infty} \boldsymbol{x}(t) \to \boldsymbol{z}(t) \tag{3.19-i}$$

since $\boldsymbol{e} = \boldsymbol{z} - \boldsymbol{x}$. Thus the adaptive laws in Eqs. 3.18-c and 3.18-d approach the plant states to the model states. These are the desired requirements for the MRAC. Hence, the controlled closed loop plant is globally asymptotically stable. Even if, the plant is unstable, it can be controlled by at least the same order stable reference model.

In the adaptive laws, $\alpha > 0$ is an arbitrary design constant referred to as the adaptive gain. The use of a larger value of the adaptive gain α led to a faster convergence of plant output to model's. The adaptive gain α *is* usually chosen by trial and error using simulations in order to achieve a good rate of convergence. Small α may result in slow convergence rate whereas large α may make the differential equations stiff and difficult to solve numerically on a digital computer (Ioannou, 1994).

For digital computations, we can convert Eq. (3.18-c) and (3.18-d) to (3.20-a) and (3.20-b), respectively:

$$\boldsymbol{F}(k+1) = \boldsymbol{F}(k) - \alpha\, \boldsymbol{x}(k)\boldsymbol{b}_m^T \boldsymbol{P}\boldsymbol{e}(k) \ , \tag{3.20-a}$$

$$g(k+1) = g(k) + \alpha\, r(k)\boldsymbol{b}_m^T \boldsymbol{P}\boldsymbol{e}(k) \ , \tag{3. 20-b}$$

which are discrete time versions and k denotes kT, k=0,1,..2 and T is the sampling period, which will be called simulation step size or adaptation period in the sequel.

There are two important matters with respect to choosing matrix $\boldsymbol{P}$. These issues are explained with related to the second order plant and reference model. The matrices used for the changes of $\boldsymbol{F}$ and g are explicitly written as

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} , \ \boldsymbol{b_m} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} , \boldsymbol{P} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \ \boldsymbol{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \ . \tag{3.21}$$

i. With these values, the adaptive laws in Eqs. 3.20-a, b are written as

$$\begin{bmatrix} F_1(k+1) \\ \mathrm{F}_2(k+1) \end{bmatrix} = \begin{bmatrix} F_1(k) \\ \mathrm{F}_2(k) \end{bmatrix} - \alpha \begin{bmatrix} x_1(k)\left[b\,e_1(k) + c\,e_2(k)\right] \\ x_2(k)\left[b\,e_1(k) + c\,e_2(k)\right] \end{bmatrix} , \tag{3.22-a}$$

$$g(k+1) = g(k) + \alpha\, r(k)\left[b\,e_1(k) + c\,e_2(k)\right] \ . \tag{3.22-b}$$

Obviously $\boldsymbol{b}_m$ has negative property for the adaptive laws. Because, except only last element, its remaining element(s) is (are) zero. For this reason and as shown in Eqs. 3.22, except last row elements, other rows elements of $\boldsymbol{P}$ are ineffective on the adaptive laws. Advise that, the absolute values of last row elements of $\boldsymbol{P}$ are not very different from each other, otherwise the effect of elements of $\boldsymbol{e}$ on the adaptive laws are different ratios (Karadeniz et al, 2004).

ii. As shown below in Eqs. 3.23-a and 3.23-b, if except diagonal elements, remaining elements of $\boldsymbol{P}$ are zero (b=0), only e_2 is effective on the adaptive laws. In these circumstances, adaptation may not occur. In order not to meet with such a result advise that, all the elements of matrix Q which is used in the Lyapunov equation (Eq. 3.16) are chosen different from zero. In fact with b=0 Eqs. 3.22-a, b reduces to

$$\begin{bmatrix} F_1(k+1) \\ \mathrm{F}_2(k+1) \end{bmatrix} = \begin{bmatrix} F_1(k) \\ \mathrm{F}_2(k) \end{bmatrix} - \alpha \begin{bmatrix} x_1(k)\big(c\, e_2(k)\big) \\ x_2(k)\big(c\, e_2(k)\big) \end{bmatrix}, \qquad \text{3.23-a}$$

$$g(k+1) = g(k) + \alpha\, r(k)\big(c\, e_2(k)\big). \qquad \text{3.23-b}$$

There are two important matters with respect to choosing of adaptation period T or simulation step size. The inverse of T is called as adaptation frequency f.

i. The selected simulation step size T must be large enough to permit the computations to be performed in real time. But the large simulation step size T may tend to result in an inferior performance in the closed loop plant and it is bad for disturbances rejection in adaptive law (CHAK at all, 1997). For this reason simulation step size T must be bigger than computation time T_c ($T > T_c$) , otherwise simulation program does not work properly and it gives error message. Computation time T_c involves the evaluation of Eqs. 3.20-a, b and it depends on hardware features. It is not measured precisely or properly and already it is not unnecessary, because Simulink gives error message automatically for ($T \le T_c$).

ii. Simulation step size T must be smaller than plant's T_{pi} and model's T_{mi} time constants ($T < T_{Pi}$ and $T < T_{mi}$). Otherwise adaptation mechanism is insufficient for perception and correction to the parameters variations of plant.

For simplicity and shortening the computation time $\boldsymbol{b}_m^T\boldsymbol{P}$ is abbreviated by $\boldsymbol{s}^T$ in the adaptive laws, i.e .,

$$\boldsymbol{s}^T = \boldsymbol{b}_m^T\boldsymbol{P}$$

Now, adaptive laws in Eqs. 3.20 a, b are defined as follow

$$\boldsymbol{F}(k+1) = \boldsymbol{F}(k) - \alpha\,\boldsymbol{x}(k)\boldsymbol{s}^T\boldsymbol{e}(k), \qquad (3.24\text{-a})$$

$$g(k+1) = g(k) + \alpha\, r(k)\boldsymbol{s}^T\boldsymbol{e}(k). \qquad (3.24\text{-b})$$

Theoretically, the initial values of adaptation gains $\boldsymbol{F}(0)$ and $g(0)$ may be any values. But, the desired performance results with a delay due to choosing the initial values very far from the steady stead values or optimal values $\boldsymbol{F}^*$ and g^*. The initial values of adaptation gains may be chosen zero or small values due to unknown optimal values. The initial values are chosen zero in this thesis.

In the next chapter, the plant which is a permanent magnet DC motor will be introduced and its MRAC is achieved as desired in this chapter.

CHAPTER 4

MRAC FOR DC MOTOR

Mathematical modeling of permanent magnet (PM) DC motor is presented. Parameters of DC motor to be controlled and order reduction of motor transfer function are investigated in this chapter. Then model reference adaptive control of the mentioned motor is presented.

4.1 MATHEMATICAL MODELLING OF PM DC MOTORS

Model of a PM DC motor is shown Figure 4.1 (Kuo, 1995).

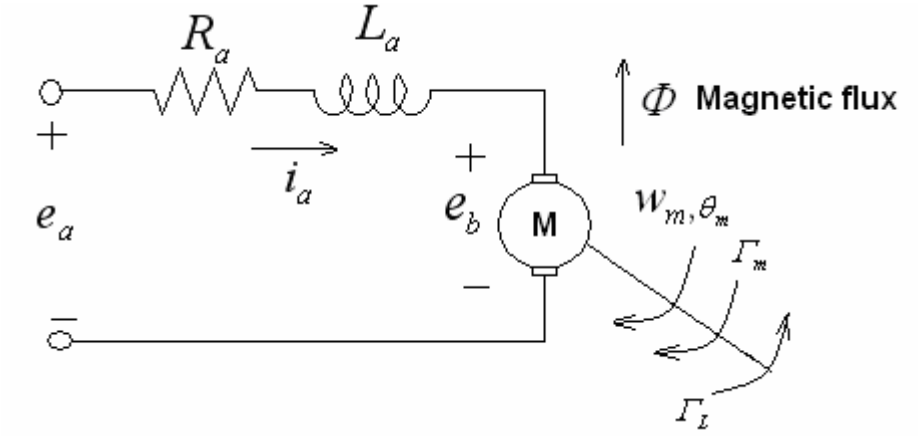

Figure 4.1 Model of a separately excited PM DC motor.

The motor variables and parameters in this figure are explained as follows:

$i_a(t)$=armature current,

R_a=armature resistance,

$e_b(t)$= back electromotive force (emf),

$T_L(t)$=load torque,

$T_m(t)$= motor torque,

$\theta_m(t)$=rotor displacement,

K_i=torque constant,

K_b=back emf constant,

L_a=armature inductance,

$e_a(t)$=applied voltage,

Φ=magnetic flux in the air gap,

$w_m(t)$=rotor angular velocity,

$J_m(t)$=rotor inertia,

B_m=viscous- friction coefficient.

The mathematical modeling of PM DC motor is given as follows:

$$\frac{di_a(t)}{d(t)} = \frac{1}{L_a}e_a(t) - \frac{R_a}{L_a}i_a(t) - \frac{1}{L_a}e_b(t), \tag{4.1}$$

$$T_m(t) = K_i i_a(t), \tag{4.2}$$

$$e_b(t) = K_b \frac{d\theta_m(t)}{dt} = K_b w_m(t), \tag{4.3}$$

$$\frac{d^2\theta_m(t)}{dt^2} = \frac{1}{J_m}T_m(t) - \frac{1}{J_m}T_L(t) - \frac{B_m}{J_m}\frac{d\theta_m(t)}{dt}. \tag{4.4}$$

By direct substitution and eliminating all the nonstate variables from Eq. (4.1) through (4.4), the state equations of dc motor are presented in vector matrix form:

$$\begin{bmatrix} \frac{di_a(t)}{d(t)} \\ \frac{dw_m(t)}{dt} \\ \frac{d\theta_m(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} & 0 \\ \frac{K_i}{J_m} & -\frac{B_m}{J_m} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} i_a \\ w_m \\ \theta_m \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} e_a(t) - \begin{bmatrix} 0 \\ \frac{1}{J_m} \\ 0 \end{bmatrix} T_L(t) \ . \qquad (4.5)$$

Notice that in this case, $T_L(t)$ is taken as second input in the state equations.

The state diagram of the system is depicted as shown in Fig. 4.2, using Eq. (4.5).

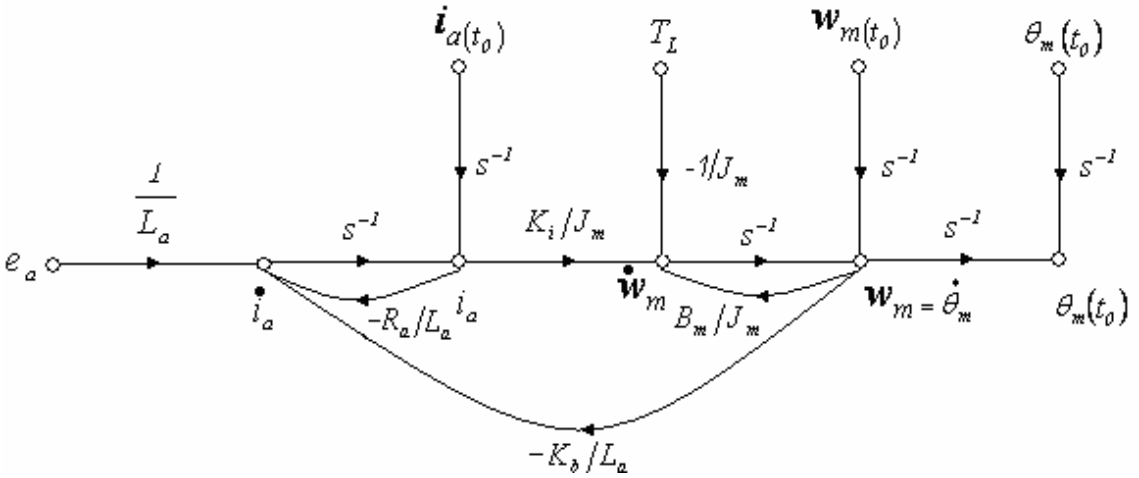

Figure 4.2 State diagram of a dc motor.

The transfer function between the motor displacement and the applied voltage is acquired from the state diagram as

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{K_i}{L_a J_m s^3 + (R_a J_m + B_m L_a)s^2 + (K_b K_i + R_a B_m)s} \qquad (4.6)$$

Here $T_L(t)$ is accepted zero. The importance of the transfer function $\Theta_m(s)/E_a(s)$ is that the dc motor is basically an integrating device between these two variables due to the transmission pole at s = 0.

Figure 4.3 displays a block-diagram representation of the dc-motor. The benefit of utilizing block diagram is that it gives a comprehensible drawing of the transfer function relations between the internal and / or external variables of the system.

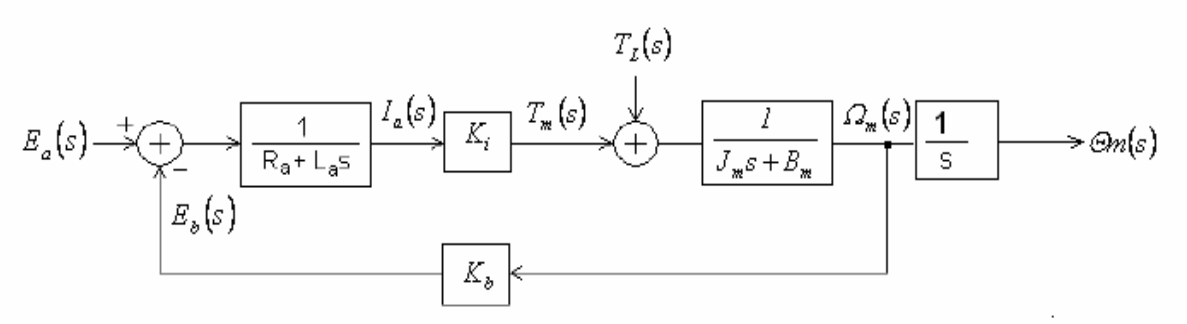

Figure 4.3 Block diagram of a dc motor system.

The values of K_b and K_i are identical if K_b is represented in V/(rad/sec) and K_i is in (Nm)/A. If K_b and K_i are replaced by K, Eq. 4.6 reduces to

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{K}{L_a J_m s^3 + (R_a J_m + B_m L_a) s^2 + (K^2 + R_a B_m) s} \qquad 4.7$$

Dividing both numerator and denominator by $L_a J_m$, Eq. 4.7 is normalized as

$$\frac{\Theta_m(s)}{E_a(s)} = \frac{K/(L_a J_m)}{s^3 + (R_a / L_a + B_m / J_m) s^2 + (K^2 + R_a B_m) s/(L_a J_m)}. \qquad 4.8$$

4.2 PARAMETERS OF DC MOTOR TO BE CONTROLLED

The knowledge of plant's order is enough for MRAC design. In our application plant's order is 3 which is seen in Eqs. 4.6-8. For simplicity and shortening the number of operation, plant's order is reduced to 2. The exact knowledge of dc motor parameters is needed for using more formal order reduction methods. For finding of the parameters, two methods are utilized which are called usual empirical method and matlab's system identification toolbox.

4.2.1 Usual Empirical Method

The transfer function between applied voltage $E_a(s)$ and rotor angular velocity $W_m(s)$ and its parameters are researched using experimental method**.** The transfer function $W_m(s)/E_a(s)$ is easily obtained by substituting of $\Theta_m(s)$ with $W_m(s)s^{-1}$ in Eq. 4.7 or in Eq. 4.8 to yield

$$\frac{W_m(s)}{E_a(s)} = \frac{K}{L_a J_m s^2 + (R_a J_m + B_m L_a)s + (K^2 + R_a B_m)}, \qquad 4.9$$

$$\frac{W_m(s)}{E_a(s)} = \frac{K/(L_a J_m)}{s^2 + (R_a / L_a + B_m / J_m)s + (K^2 + R_a B_m)/(L_a J_m)}, \qquad 4.10$$

respectively.

R_a and L_a are easily measured by RLC meter to be $R_a = 15.36\,\Omega$, $L_a = 0.42$ mH.
Other parameters are found by applying the following experimental scheme[2] shown in Fig. 4.4. Where, E_a is applied voltage, S is on-off switch, A is ammeter, M1 is controlled motor, E is encoder, w_m is rotor angular velocity (rad/s). Encoder output which is connected to the data acquisition board is displacement. The derivative of the displacement which is called angular velocity is realized by matlab's simulink.

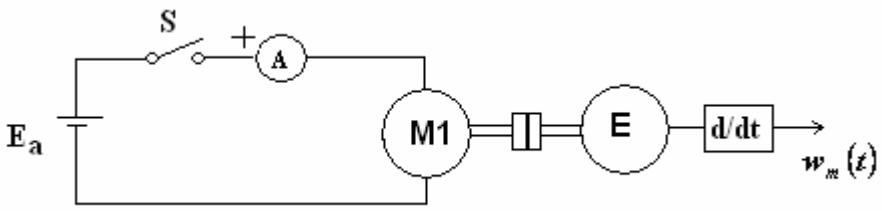

Figure 4.4 Connection scheme for empirical method.

[2] For a detailed version of the experimental setup refer to Fig. 4.13 where many of the subsytems are made functionless to obtain experimental test implied by Fig. 4.4.

The applied voltage E_a is 8.00 V. After the switch is closed, steady-stead armature current is measured as I_a =120 mA and velocity alteration becomes as in Fig. 4.5. Steady-stead velocity is 668 rad/s. Dc motor has two time constants. One of them is very large with respect to the other. The big time constant is observable, the smaller time constant is not observable on this figure, because its mode goes rapidly to zero as soon as motor starts. As can be obtained in Fig. 4.5, motor reaches 63 % of its steady-stead velocity value at τ_1 = 0.0638 second where τ_1 is the larger time constant of motor. Hence, (s+τ_1^{-1}) is one the factor of the denominator polynomial of the motor transfer function $W_m(s)/E_a(s)$. The smaller time constant or pole is to be found by computing.

Under steady-state conditions, i.e. for constant i_a and e_b, Eq. 4.1 can be written as

$$E_b = E_a - I_a R_a \ , \qquad 4.11$$

where, E_a = 8.00 V, I_a = 120 mA and R_a = 15.36 Ω. (Although the result are valid with in 2 digits due to measurement errors in the value of L_a, and 3 digits of I_a ,W_m, in order not to cause the additional accumulative errors due to computation at least 4 digits are carried in the following.) This equation yields

$E_b = K_b W_m = 8.00 - 0.120 \times 15.36 = 6.157$ V. Since, W_m is measured to be 668 rad/s the torque or back emf constant is found to be

$$K_b = K_i = K = \frac{E_b}{W_m} = \frac{6.157}{668} = 92.17 \times 10^{-4} \text{ Vs/r} . \qquad 4.12$$

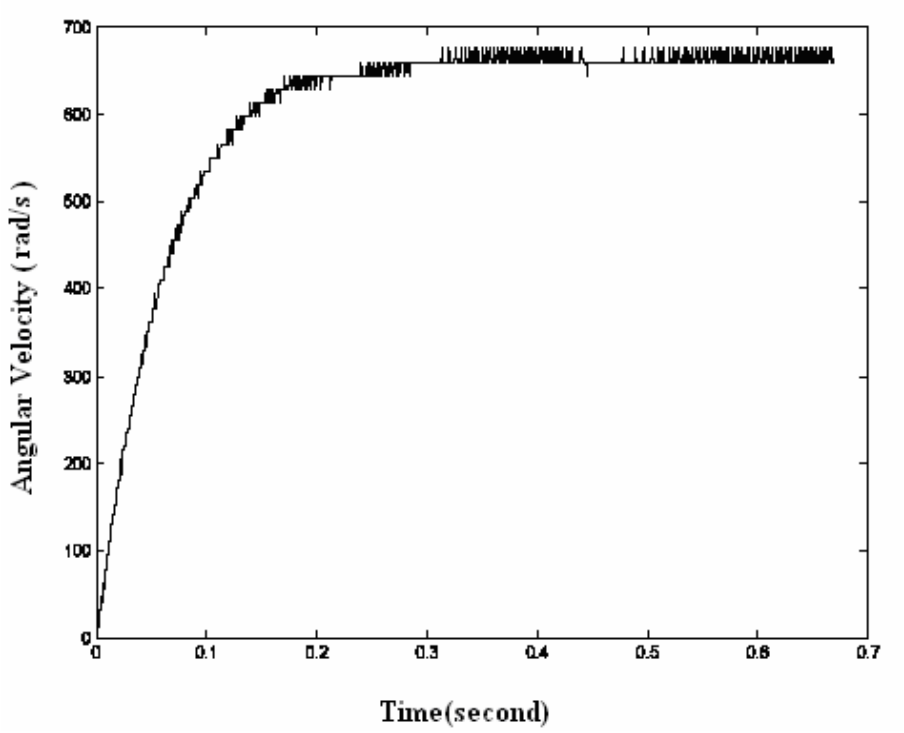

Figure 4.5 Velocity alteration of motor

For computation of B_m, consider the dc gain

$$\frac{W_m}{E_a} = \frac{K}{K^2 + R_a B_m} \qquad 4.13$$

computed from Eq. 4.10 replacing s by zero with W_m=668r/s, E_a=8.00V, $K = 92.17 \times 10^{-4}$ Vs/r, $R_a = 15.36\ \Omega$; this equation yields $B_m = 1.656 \times 10^{-6}$ Nms/r.

For computation of J_m, the denominator polynomial of the transfer function in Eq. 4.10 is divided by (s+τ_1^{-1}) and the remainder is equated to zero. This operation leads to

$$\tau_1 \left(K^2 + R_a B_m\right) = J_m \left(R_a - L_a/\tau_1\right) + L_a B_m . \qquad 4.14$$

With the measured values of R_a, L_a and the already computed values of τ_1, K, B_m ($R_a = 15.36\Omega$, L_a=0.42 mH), τ_1 = 0.0638s, $K = 92.17 \times 10^{-4}$ Vs/r, $B_m = 1.656 \times 10^{-6}$ Nms/r) this equation yields $J_m = 4.587 \times 10^{-7}$ Kg m^2.

Thus all parameters of motor are found by a resistance measurement (R_a), an inductance measurement (L_a), the steady-stead test (W_m) and the transient test (τ_1). In fact, the last two tests are combined in the step response of the motor shown in Fig. 4.5. Hence, the transfer function H_{EMP} which is found by empirical method between applied voltage E_a and displacement Θ_m is obtained by inserting the found parameters in Eq. 4.8. The result is as in Eq. (4.15), where, the second time constant τ_2 is 2.735×10^{-5} s.

$$H_{EMP}=\frac{\Theta(s)}{E_a(s)}=\frac{4.784\times10^7}{s^3+3.656\times10^4\,s^2+5.729\times10^5\,s}=\frac{4.784\times10^7}{s(s+3.656\times10^4)(s+15.67)}$$

$$=\frac{83.51}{s\left(1+2.735\times10^{-5}\,s\right)\left(1+0.0638\,s\right)} \qquad 4.15$$

4.2.2 Parameter Estimation with Matlab' s System Identification Toolbox

System Identification Toolbox of Matlab builds mathematical models from measured input-output data. The transfer function of plant is estimated using Prediction Error Method from the measured dataset. In our application, measured dataset is acquired from executing the simulink scheme shown in Fig.4.6.

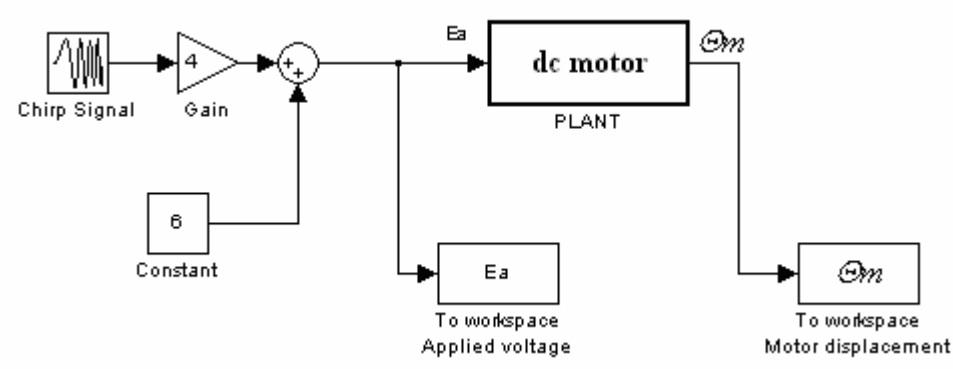

Figure 4.6 Simulink scheme for getting measured input-output data

The motor input signal E_a is a swept sinusoid (a chirp signal) whose frequency range is selected as between 0.05 and 10 Hz. The bottom level of the input signal E_a is chosen as 2 Volt, because the motor has dead zone between -1 and 1 V which produce bad effect for estimation of proper transfer function. Upper level of E_a selected as 10 V which is the

allowed maximum input voltage of motor. The output signal of motor is the shaft displacement Θ_m which is obtained from output of the encoder. These input output data are saved to matlab's workspace while the simulink program is executing. For estimation operation, Ident which is a graphical user interface to the System Identification Toolbox is started by writing to command line '*ident*'. The measured dataset import to Ident.

The transfer functions are estimated within certain classes of candidate descriptions (model or template transfer functions). We select model transfer function $K/[s(1+s\tau_1)(1+s\tau_2)]$ which is shown in left pane of Fig. 4.7. It contains one integral element, two poles and one gain. At first estimation, the fit rate between measured output and simulated output is very high (99.9 %) but τ_1, τ_2 and K are dissimilar from those in Eq.4.15. Before the second estimation, τ_1, τ_2 which are put in to the template transfer function are accepted as known values from Eq. 4.15. As shown Fig.4.7, K is found as 84.23 whilst the experimental value is 83.51. Hence, the transfer function H_{IDE3} which is found by the identification toolbox becomes as in Eq. 4.16. As shown left pane of Fig.4.8, measured output and simulated model output are seem one line. The fit rate between them is (99.9 %). It is a very good estimate.

$$H_{IDE3} = \frac{84.23}{s\left(1+2.735\times10^{-5}s\right)\left(1+0.0638s\right)} = \frac{4.827\times10^{7}}{s^3+3.656\times10^{4}\,s^2+5.729\times10^{5}\,s}$$

$$= \frac{4.827\times10^{7}}{s\,(s+3.656\times10^{4})\,(s+15.67)} \qquad 4.16$$

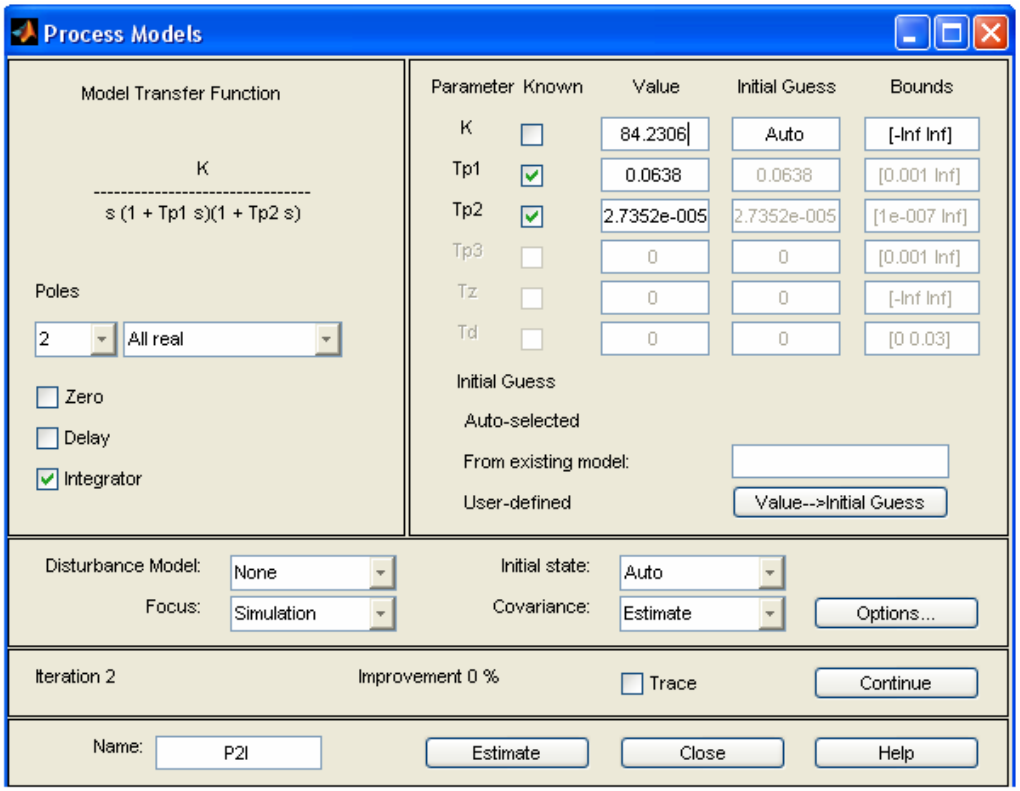

Figure 4.7 Parameter identification window related with third order estimation.

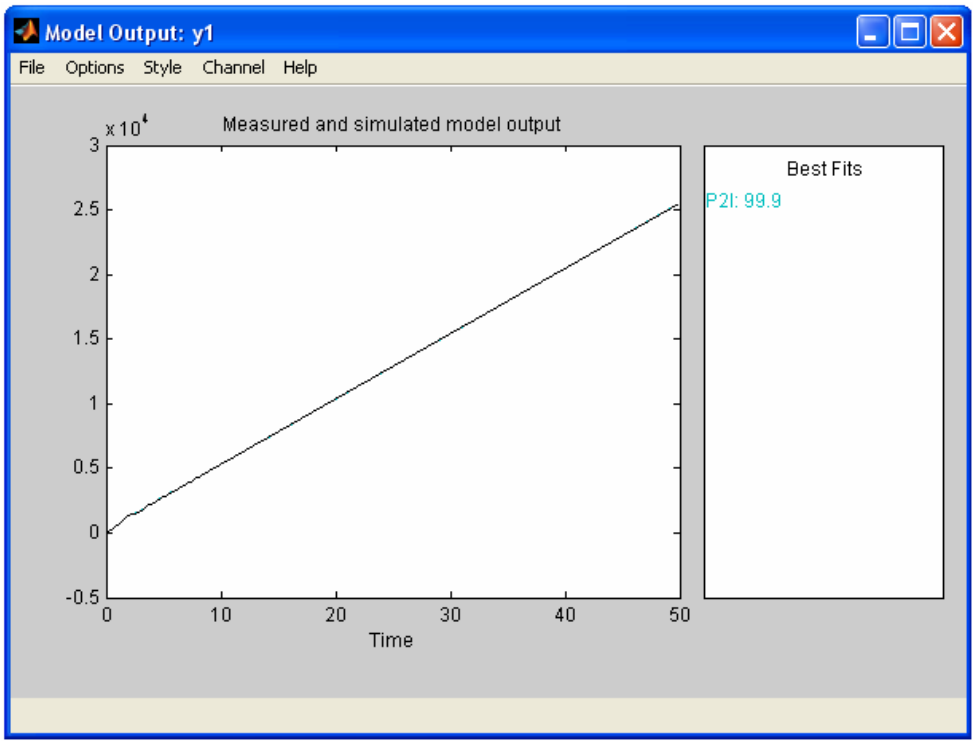

Figure 4.8 Comparison of measured and simulated outputs related with third order estimation.

To check the accuracy of the experimental results for the motor parameters, we now want to find motor parameters using Eq. 4.16. So Eq. 4.16 and Eq. 4.8 are equalized each other. Three equations which are shown below are obtained from this.

$$K/(L_a J_m) = 4.827\times10^7 \ \ (\mathrm{r/s})^3/\mathrm{V} \qquad 4.17$$

$$R_a / L_a + B_m / J_m = 3.656\times10^4 \ \ \mathrm{r/s}, \qquad 4.18$$

$$\left(K^2 + R_a B_m\right)/(L_a J_m) = 5.729\times10^5 \ \ (\mathrm{r/s})^2 \qquad 4.19$$

There are 5 unknown parameters in these equations, for solution of the equations; we must accept that, the values of whichever at least two are known. As mentioned in Section 4.2.1 that, R_a L_a and K are known values with high confidence.

Suppose that, $R_a = 15.36\,\Omega$ and $K = 92.17\times10^{-4}$ Vs/r are known and B_m, J_m, L_a are desired parameters. The solution of the equations for the remaining 3 parameters are

$B_m = 1.591\times10^{-6}$ Nms/r,

$J_m = 4.534\times10^{-7}$ Kg m^2

$L_a = 42.01\times10^{-5}$ H.

The consistency of these results with the experimental ones are correct within errors 96.07%, 98.84%, and 99.97%, respectively.

For the second case assume $L_a = 42\times10^{-5}$ H and $K = 92.17\times10^{-4}$ Vs/r are precisely determined experimentally and B_m, J_m, R_a are desired parameters computed from the result of Identification Toolbox. The solution of equations in 4.17-19 for the remaining variables B_m, J_m, R_a become

$B_m = 1.592\times10^{-6}$ Nms/r,

$J_m = 4.546\times10^{-7}$ Kg m^2,

$R_a = 15.35\,\Omega$.

These are consistent with the experimental results within coherency 96.13%, 99.11% and 99.93%, respectively.

Consequently, these results based on the System Identification Toolbox are in good agreement with experimental ones of the previous section.

4.3 APPROXIMATION OF HIGH ORDER SYSTEMS BY LOW ORDER SYSTEM

A high order system often contains less important poles that have little effect on the system response. Thus, given a high order system, it is desirable to find a low order approximating system, so that, the number of operation is reduced for the solution of control problem. Generally, in many applications which are utilizing dc motor, plant's order is accepted 2 due to neglecting of small L_a. In addition to negligence of L_a, we will offer extra two methods for order reduction.

4.3.1 Order Reduction with Negligence of L_a

In this method, L_a is accepted zero. Replacement of $L_a = 0$ in Eq. 4.7 and using the first set of parameter values obtained by System Identification Toolbox found in Section 4.2.2, namely

$R_a = 15.36\,\Omega$,

$K = 92.17\times10^{-4}$ Vs/r,

$L_a = 42.01\times10^{-5}\,\mathrm{H} \cong 0$,

$B_m = 1.591\times10^{-6}$ Nms/r,

$J_m = 4.534\times10^{-7}$ (Kg m^2),

the reduced transfer function

$$H_{L\,neg} = \frac{1323}{s\,(s+15.71)} \qquad 4.20$$

is obtained.

4.3.2 Order Reduction with Matlab's "MODRED" and "BALREAL" Commands

The order of the transfer function in Eq.4.16 is reduced to 2 using "modred" (model reduction) and "balreal" (balanced realization) commands[3]. Reduced transfer function $H_{mod\ red}$ is

$$H_{mod\ red} = \frac{1319}{\mathrm{s}\,(\mathrm{s}+15.65)} \qquad 4.21$$

which is almost the same within an error less than 0.2 % in the coefficients as in Eq. 4.20 obtained by simply replacing L_a by zero.

4.3.3 Order Reduction with Matlab's System Identification Toolbox

Second order transfer function of motor is estimated from measured dataset which is the same dataset used in estimation of Eq.4.16. This time, we select template transfer function $K/[s(1+s\tau_1)]$ as shown in left pane of Fig.4.9. This transfer function contains one integral element, one pole and one gain. Hence, the transfer function $H_{IDE\,2}$ which is predicted as second order becomes

$$H_{IDE\,2} = \frac{1319}{\mathrm{s}\,(\mathrm{s}+15.66)} \qquad 4.22$$

This result is also in good coherence with those in Eqs. 4.20 and 4.21. In fact, having the most similarity to the remaining two others, this model will be taken as reference in the sequel when discussing the MRAC.

As shown in the left pane of Fig. 4.10, measured output and simulated model output are coincident. The fit rate between them is 99.9 % which is the same fit rate of Eq. 4.16.

[3] The matlab program code with concerning these commands is placed in to the C.2

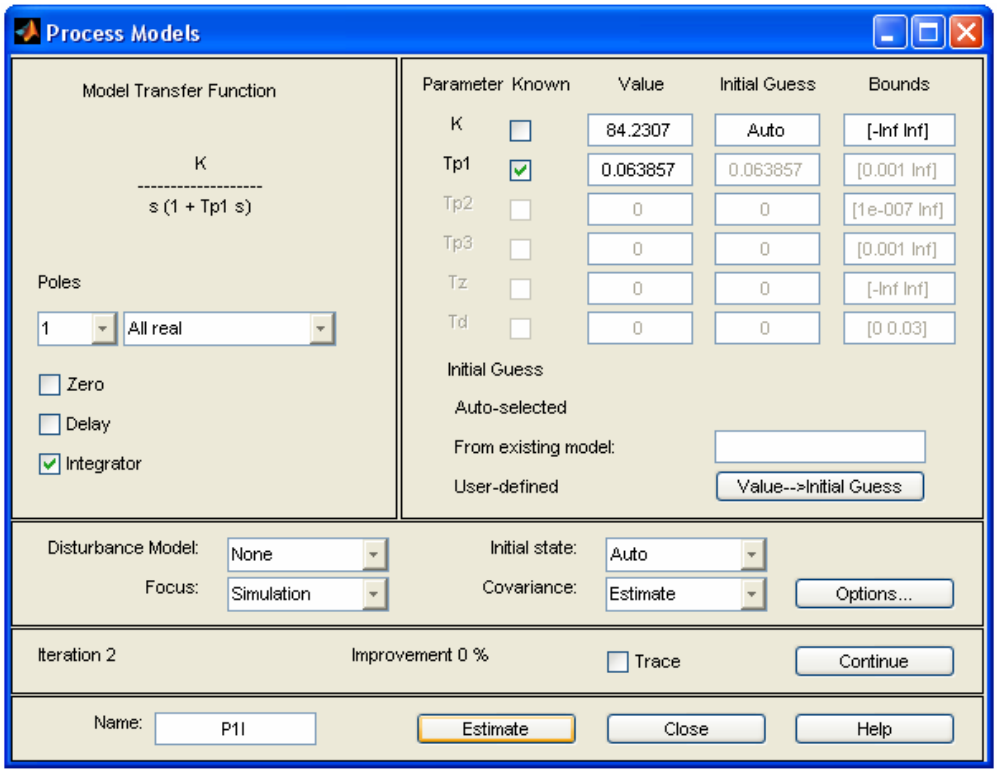

Figure 4.9 Parameter identification window related with second order estimation.

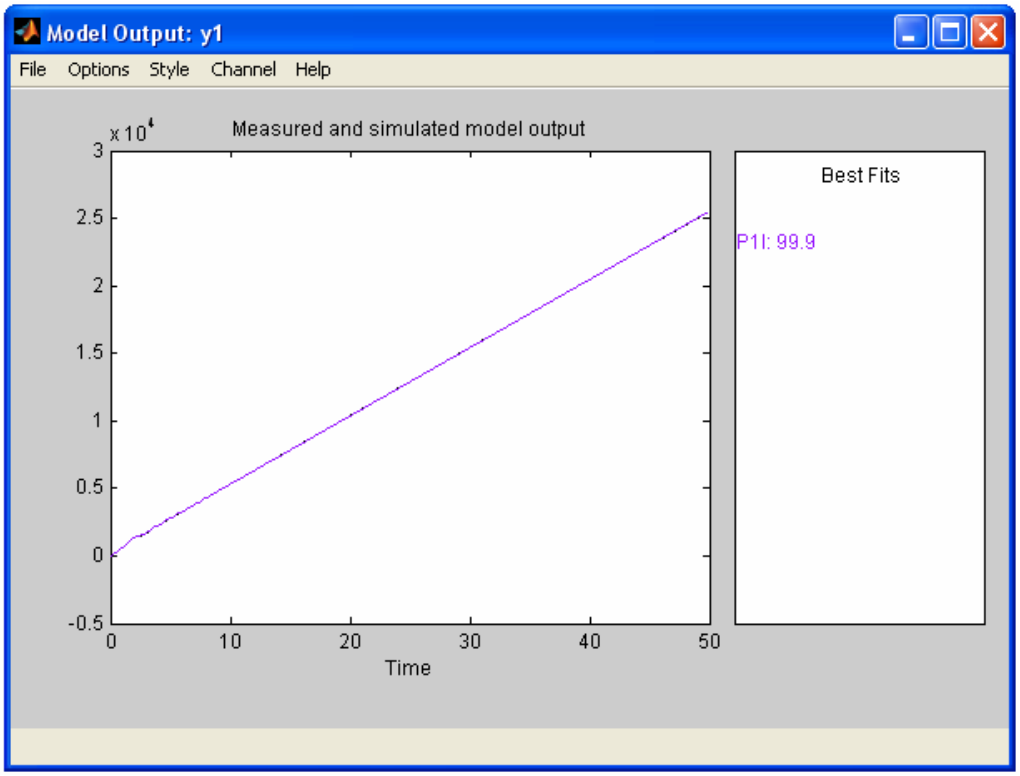

Figure 4.10 Comparison of measured and simulated outputs related with second order estimation.

4.3.4 Comparison Between Third Order Transfer Function and Second Order Transfer Function

In the previous section, it has already been concluded that the results of different reduction techniques are in good agreement. The transfer functions in Eqs. 4.16, 4.20, 4.21, 4.22 are now compared with each other by plotting bode diagrams. All bode diagrams are shown in Fig. 4.11.

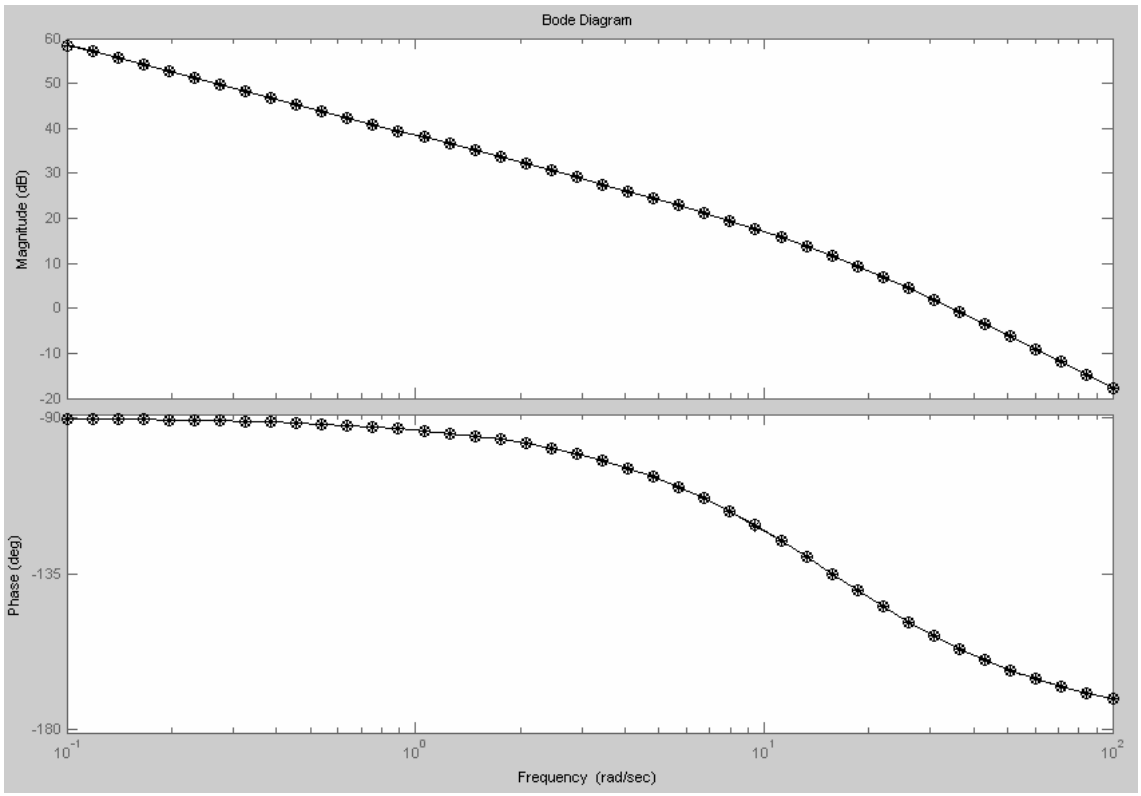

Fig. 4.11 Bode Diagrams of Transfer Functions

In this figure bode diagrams of the transfer functions in Eqs. 4.16, 4.20, 4.21, 4.22 are remarked with . (point), o (circle), x (multiplication), + (plus) symbols, respectively. All of the bode curves concerning third and second order transfer functions seem identical over a 3 decade frequency range strating from 0.1 r/s and extending to 100r/s. Hence, the order of dc motor's transfer function $\Theta_m(s)/E_a(s)$ may be well accepted as second order, and all the reduced second order transfer functions are almost equally well.

4.4 MRAC OF PM DC MOTORS

Since, the motor can be accepted as second order, the order of reference model and the number of states and its feedback gains of dc motor should be two accordingly. The principle MRAC scheme of a dc motor displacement is shown in Fig.4.12. Where, $\theta_r(t)$ is reference signal input which is chosen as square wave. $e_a(t)$ is the input voltage of motor. The displacement $\theta(t)$ is the first state of motor and it is assigned to be output. $w_m(t)$ is the second state or velocity of the motor which is gotten by first derivative of displacement. F_1 and F_2 are feedback gains of first and second states. g is feed forward gain of reference signal. e(t) is identified as difference or error signal between reference model output and dc motor output. Adaptation mechanism whose input are first state, second state, reference signal, error signal and its derivative adjust feedback gains F_1 , F_2 and feed forward gain g. The aim of the adaptation is that, dc motor output converges to reference model output and error signal goes to zero.

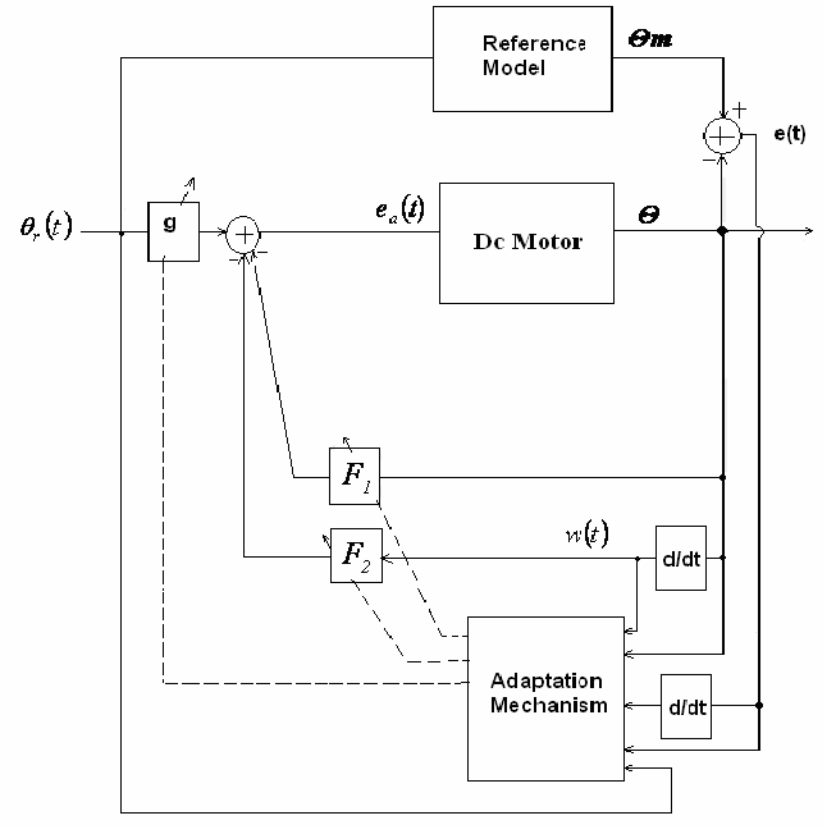

Figure 4.12 The principle MRAC scheme of a dc motor displacement.

4.4.1 Experimental Setup for MRAC

Physical scheme of MRAC of dc motor displacement is shown in Fig. 4.13. Where, M1 is the controlled motor which is a permanent magnet dc motor whose control voltage is got from female connector X1 of data acquisition board. E is dual channel encoder which measures displacement of motor shaft. The phases of the two-output (A and B) pulse trains are 90 degrees apart electrically. Each output produce 100 square wave per rotation. Encoder outputs are connected to female connector X2 of data acquisition board. M_D which is identical with M1 is permanent magnet dc motor. It acts as load to for M1 and applies disturbance effect for displacement. Disturbance torque or load of M1 (T_L) is adjusted by the armature current I_D of M_D.

R_{ext} is $200\,\Omega$ external resistance which provides parameter variations for armature resistance so it applies disturbance effect for displacement. The maximum output voltage and maximum output current of motor driver are limited 10 V and 0.5 A, respectively. The shafts of M1, E and M_D are coupled to each other for experiment[4].

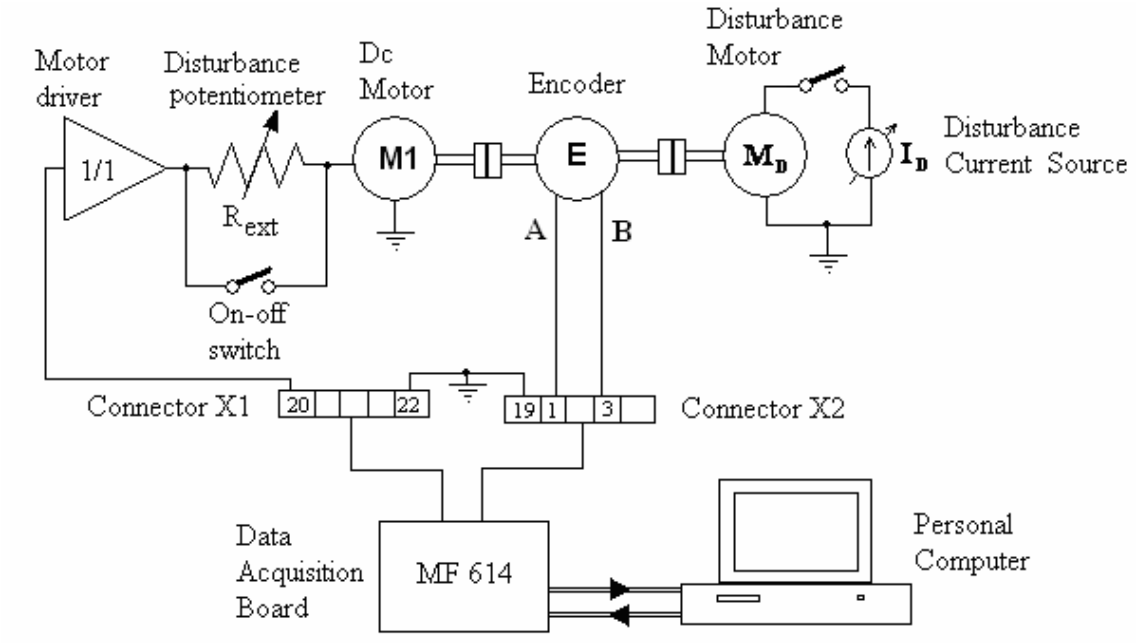

Figure 4.13 Physical scheme of MRAC of a dc motor displacement.

[4] The features of appliances and program codes are in appendix B and appendix C respectively.

4.4.2 Review of Adaptive Laws for the Control of Tested Motors

The second order mathematical model in Eq. 4.22 is assumed as reference for the plant of this section. Adapting the notation in Fig. 4.12, we write this transfer function again for convenience

$$\frac{\Theta}{1319\,E_a} = \frac{1}{s(s+15.66)} \qquad 4.23$$

Direct realization procedure yields the following steps:

$$\frac{\Theta}{1319\,E_a} = \frac{1}{s^2+15.66s} = \frac{s^{-2}}{1+15.66\,s^{-1}} = \frac{s^{-2}F}{F+15.66\,s^{-1}F} \quad . \qquad 4.24\text{-a}$$

Equating the numerators and denominators, we obtain

$$\Theta = s^{-2}F\,, \qquad 4.24\text{-b}$$

$$F = 1319E_a - 15.66\,s^{-1}\,F\,, \qquad 4.24\text{-c}$$

respectively. The following state transition signal flow graph in Fig. 4.14 follows directly.

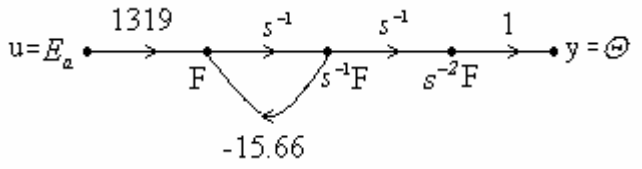

Fig. 4.14 State transition signal flow graph of the motor model.

Designing Θ and $w = d\Theta/dt$ as the first and second state equations, we obtain the controllable state equations

$$\frac{d}{dt}\begin{bmatrix}\Theta\\ w\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -15.66\end{bmatrix}\begin{bmatrix}\Theta\\ w\end{bmatrix} + \begin{bmatrix}0\\ 1319\end{bmatrix}e_a\,, \qquad 4.25\text{-a}$$

$$\Theta = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\Theta\\ w\end{bmatrix} \quad . \qquad 4.25\text{-b}$$

These equations constituting the plant model correspond to Eqs. 3.2-a and 3.2-b respectively. Hence

$$A = \begin{bmatrix}0 & 1\\ 0 & -15.66\end{bmatrix},\; B = \begin{bmatrix}0\\ 1319\end{bmatrix}.$$

Feedback control law in Eq. 3.4 becomes

$$e_a = g\,r - \begin{bmatrix} F_1 & F_2 \end{bmatrix} \begin{bmatrix} \Theta \\ w \end{bmatrix}. \qquad 4.26$$

For the reference model, we assume a prototype second order transfer function with undamped natural frequency w_n and damping coefficient ξ (Kuo, B. C., 1995). Hence,

$$\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{{w_n}^2}{s^2 + 2\xi w_n s + {w_n}^2} \quad . \qquad 4.27$$

Following Eqs. 3.6-a, 3.6-b, 3.6-c and 3.6-d, we have

$$\frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -w_n^2 & -2\xi w_n \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ w_n^2 \end{bmatrix}\Theta_r, \qquad 4.28$$

$$\Theta_m = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \qquad 4.29$$

$$A_m = \begin{bmatrix} 0 & 1 \\ -w_n^2 & -2\xi w_n \end{bmatrix}, \quad B_m = \begin{bmatrix} 0 \\ w_n^2 \end{bmatrix}, \qquad 4.30$$

for the reference model.

Finally adaptation laws are obtained explicitly using Eq. 3.20-a and b,

$$\begin{bmatrix} F_1(k+1) \\ \mathrm{F}_2(k+1) \end{bmatrix} = \begin{bmatrix} F_1(k) \\ \mathrm{F}_2(k) \end{bmatrix} - \alpha \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}\begin{bmatrix} 0 & w_n^2 \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}\begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix}, \qquad 4.31$$

$$g(k+1) = g(k) + \alpha\,\Theta_r(k)\begin{bmatrix} 0 & w_n^2 \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}\begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix}. \qquad 4.32$$

The explicit form of the Lyapunov Equation in (3.16) becomes

$$\begin{bmatrix} 0 & -w_n^2 \\ 1 & -2\xi w_n \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -w_n^2 & -2\xi w_n \end{bmatrix} = -\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \qquad 4.33$$

We find it convenient to remind that the rate of decrease of the Lyapunov function V is determined by Q due to Eq. 3.19 i.e,

$$\dot{\boldsymbol{V}} = -\boldsymbol{e}^T \boldsymbol{Q}\, \boldsymbol{e}\,.$$

The error decays to zero by the time constants of the reference model if $\boldsymbol{\Phi}_{\boldsymbol{F}}$ and Φ_g in Eqs. 3.13 become zero shortly so that this equation reduces to $\dot{\boldsymbol{e}} = \boldsymbol{A}_{\boldsymbol{m}} \boldsymbol{e}$.

On the base of the specific MRAC equations associated with the plant, the reference model the MRAC method given in this chapter, the obtained simulation result will be presented in the next chapter.

CHAPTER 5

SIMULATION RESULTS

In this chapter, we investigate the performance of adaptive tracking for various adaptive gains α, positive definite matrices *P*, simulation step sizes *T,* reference models and various disturbance effects.

5.1 EFFECTS OF ADAPTIVE GAIN α ON THE PERFORMANCE

We investigate in this section, the effect of adaptive gain α on the adaptive tracking performance for a proper positive definite matrix *P*, for fixed simulation step size *T* and a chosen reference model. We can suggest a reference model having the transfer function $\frac{\Theta_m(s)}{\Theta_r(s)} = \frac{{w_n}^2}{s^2 + 2\zeta w_n s + {w_n}^2}$ via choosing proper values for the ξ and w_n parameters. As shown in Eq. 5.1 and Eq. 5.2, percent maximum overshoot and quality factor of the denominator polynomial is adjusted by only ζ, as

$$\text{Percent maximum overshoot} = 100\, e^{-\pi\zeta/\sqrt{1-\zeta^2}}, \qquad 5.1$$

$$\text{Quality factor of denominator polynomial} = 1/(2\zeta). \qquad 5.2$$

The time delay t_d, rise time t_r and settling time t_s are approximately adjusted by both ς and w_n (Kuo, 1995) as in the following formulas, respectively;

$$t_d \cong (1.1 + 0.125\zeta + 0.469\zeta^2)/w_n, \qquad 5.3$$

$$t_r \cong (1 - 0.4167\zeta + 2.917\zeta^2)/w_n, \qquad 5.4$$

$t_s \cong 3.2/(\zeta w_n)$ for $0 < \zeta < 0.69$, 5.5-a

$t_s \cong (4.5\zeta)/w_n$ for $\zeta > 0.69$. 5.5-b

We choose ζ =1 and w_n = 4 for the reference model. We get the following results by using the formulas 5.1 through 5.5;

Percent maximum overshoot = 0,

Quality factor of denominator polynomial = 0.5 < 1,

$t_d \cong 0.4235$ s,

$t_r \cong 0.875$ s,

$t_s \cong 1.125$ s.

From Eq. 4.30, $\boldsymbol{A_m}$ and $\boldsymbol{B_m}$ are

$$\boldsymbol{A_m} = \begin{bmatrix} 0 & 1 \\ -16 & -8 \end{bmatrix}, \quad \boldsymbol{B_m} = \begin{bmatrix} 0 \\ 16 \end{bmatrix}.$$

Q is chosen as $\boldsymbol{Q} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$.

$\boldsymbol{P}$ is solved from the Lyapunov Equation 4.33 which becomes

$$\begin{bmatrix} 0 & -16 \\ 1 & -8 \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -16 & -8 \end{bmatrix} = -\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix},$$

Its solution for $\boldsymbol{P}$ is

$$\boldsymbol{P} = \begin{bmatrix} 0.625 & 0.0625 \\ 0.0625 & 0.0703 \end{bmatrix}.$$

$\boldsymbol{P}$ satisfies criterions that are mentioned at the end of Section 3.2. Namely, all elements of $\boldsymbol{P}$ are different from zero and the elements of the bottom row be close each other ($b \cong c$).

From Eqs. 4.31 and 4.32, adaptive laws are

$$\begin{bmatrix} F_1(k+1) \\ \mathrm{F}_2(k+1) \end{bmatrix} = \begin{bmatrix} F_1(k) \\ \mathrm{F}_2(k) \end{bmatrix} - \alpha \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \begin{bmatrix} 0 & 16 \end{bmatrix} \begin{bmatrix} 0.625 & 0.0625 \\ 0.0625 & 0.0703 \end{bmatrix} \begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix},$$

$$g(k+1) = g(k) + \alpha \Theta_r(k) \begin{bmatrix} 0 & 16 \end{bmatrix} \begin{bmatrix} 0.625 & 0.0625 \\ 0.0625 & 0.0703 \end{bmatrix} \begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix}.$$

As used in Eqs. 3.24a-b, $\boldsymbol{b}_m^T \boldsymbol{P}$ is abbreviated by $\boldsymbol{s}^T = \begin{bmatrix} 1 & 1.125 \end{bmatrix}$.

So adaptive laws are written by simple way

$$\begin{bmatrix} F_1(k+1) \\ \mathrm{F}_2(k+1) \end{bmatrix} = \begin{bmatrix} F_1(k) \\ \mathrm{F}_2(k) \end{bmatrix} - \alpha \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \begin{bmatrix} 1 & 1.125 \end{bmatrix} \begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix},$$

$$g(k+1) = g(k) + \alpha \Theta_r(k) \begin{bmatrix} 1 & 1.125 \end{bmatrix} \begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix}.$$

Adaptation period T is chosen as 1 ms. The time constant of the plant (Eq. 4.22) is $T_p = 1/15.66 = 6.38 \times 10^{-2}\, s \cong 64ms$ and the time constants of model are $T_{m1} = T_{m2}$=250 ms. Simulation step size T is sufficiently smaller than the plant's and model's time constants ($T<T_{pi}$ *and* $T<T_{mi}$). It satisfies criterions that mentioned at the end of Section 3.2.

Simulation step size T is bigger than the computation time T_c because computer is forced and gives error message due to using simulation step size T smaller than 1ms, i.e; $T_c \approx 1$ ms. In fact T=1 ms is almost the minimum allowable value and the effects of other values are also investigated in the sequel.

Reference signal is a square wave, its frequency is chosen 0.1 Hz (period is 10 s) and its amplitude is chosen between $\pi/2$ and π in all simulations.

As mentioned at the end of Section 3.2, adaptive gain α will be chosen by trial and error using simulations in order to achieve a good rate of convergence. Simulation results for the rotor angular position are shown in Figures 5.1, 5.2, 5.3 for adaptive gains $\alpha = 10^{-4}, \alpha = 10^{-3}, \alpha = 10^{-2}$ respectively. All the figures indicate satisfactory convergence of the plant output to that of the model. But the use of a larger value of the adaptive gain α led to a faster convergence of plant output to model's. The biggest adaptive

gain is chosen as $\alpha = 10^{-2}$ because adaptive laws become stiff and difficult to solve numerically on the computer and simulink gives application error for $\alpha > 10^{-2}$. The solid lines show model output, dotted lines show plant output, solid-dotted lines if exist show error signal in all simulations.

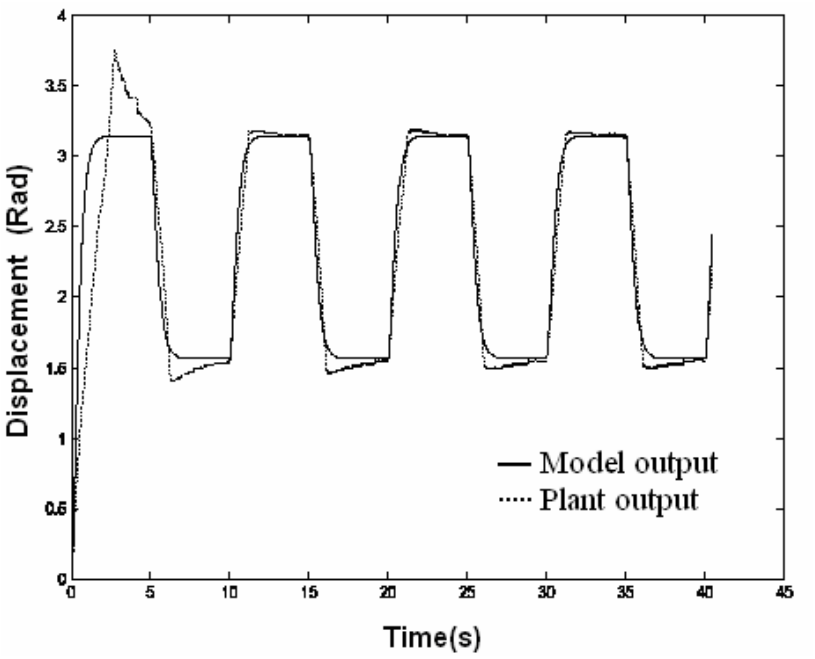

Figure 5.1 Simulation results for $\alpha = 10^{-4}$, $b \cong c$, T=1 ms

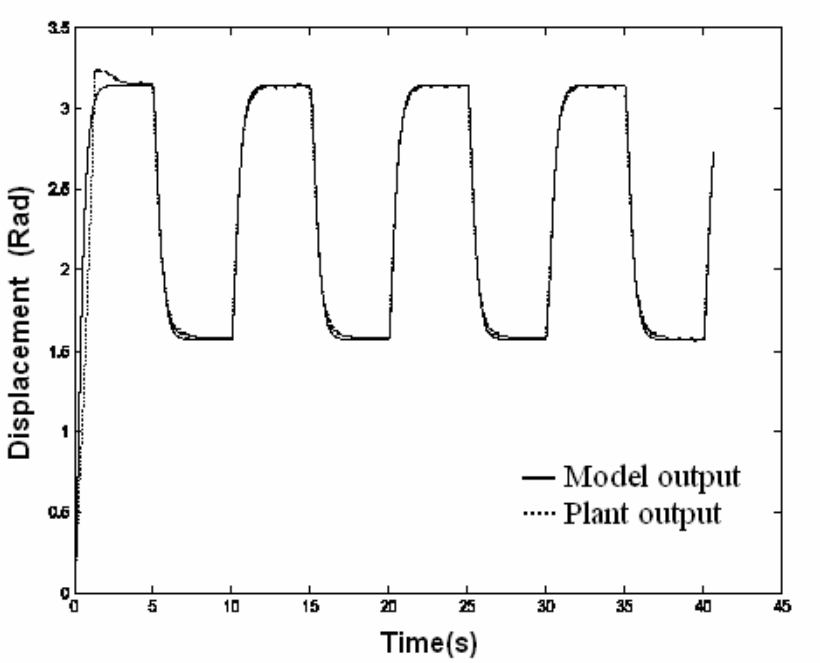

Figure 5.2 Simulation results for $\alpha = 10^{-3}$, $b \cong c$, T=1 ms

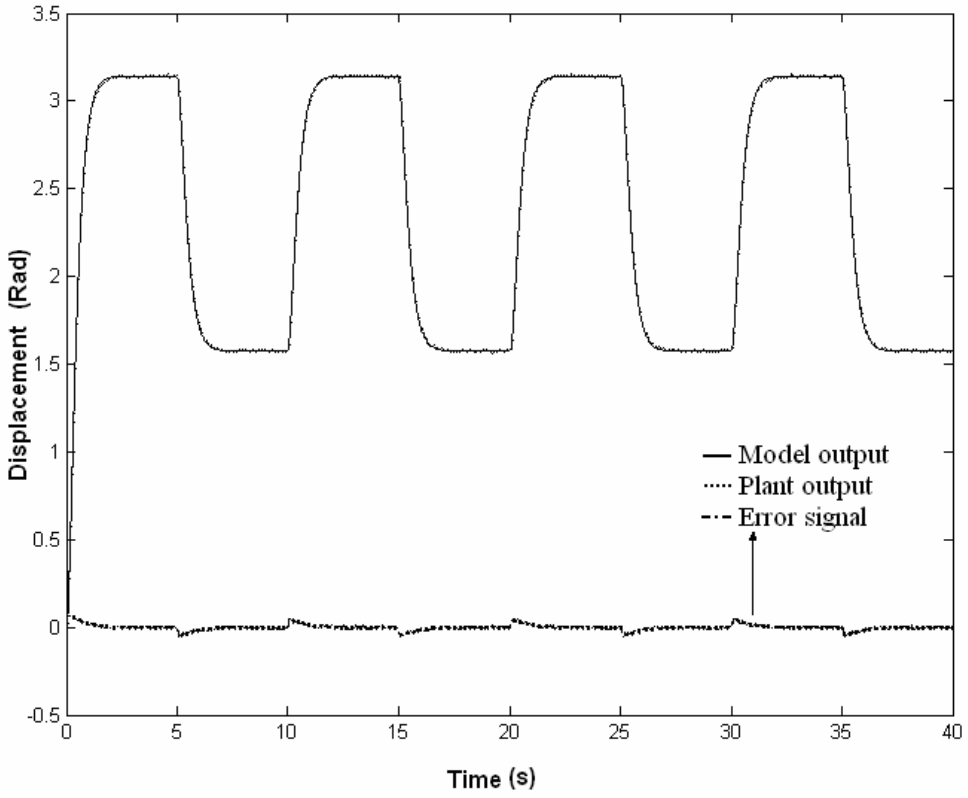

Figure 5.3 Simulation results for $\alpha = 10^{-2}$, $b \cong c$, T=1 ms.

5.2 EFFECTS OF POSITIVE DEFINITE MATRIX P ON THE PERFORMANCE

Now, we investigate how adaptive tracking performance is affected, if $\boldsymbol{P}$ matrix is chosen in improper form. We consider three different $\boldsymbol{P}$ matrices whose bottom elements are not close each other for the first two cases, the first bottom element is equal to zero for the last case.

i. $\boldsymbol{P} = \begin{bmatrix} 2.126 & 0.3125 \\ 0.3125 & 0.0391 \end{bmatrix}$ for $\boldsymbol{Q} = \begin{bmatrix} 10 & 1 \\ 1 & 0.001 \end{bmatrix}$,

As seen in Eqs. 3.22-a and 3.22-b, the first bottom element (b) of $\boldsymbol{P}$ is the multiplier of e_1 ($z_1 - \Theta$) and the second bottom element (c) of $\boldsymbol{P}$ is the multiplier of e_2 $(z_2 - w)$. This situation may be stated as, e_1 is 0.3125/0.0391=8 times more effective than e_2 on the adaptive laws. The simulation results are shown in Fig. 5.4 with respect to this situation. The error signal e_1 changes between +0.15 rad and -0.15 rad. The error signal in Fig. 5.4 is 3 times bigger than in Fig. 5.3. Shortly, the case of ($b >> c$) is less successful than ($b \cong c$).

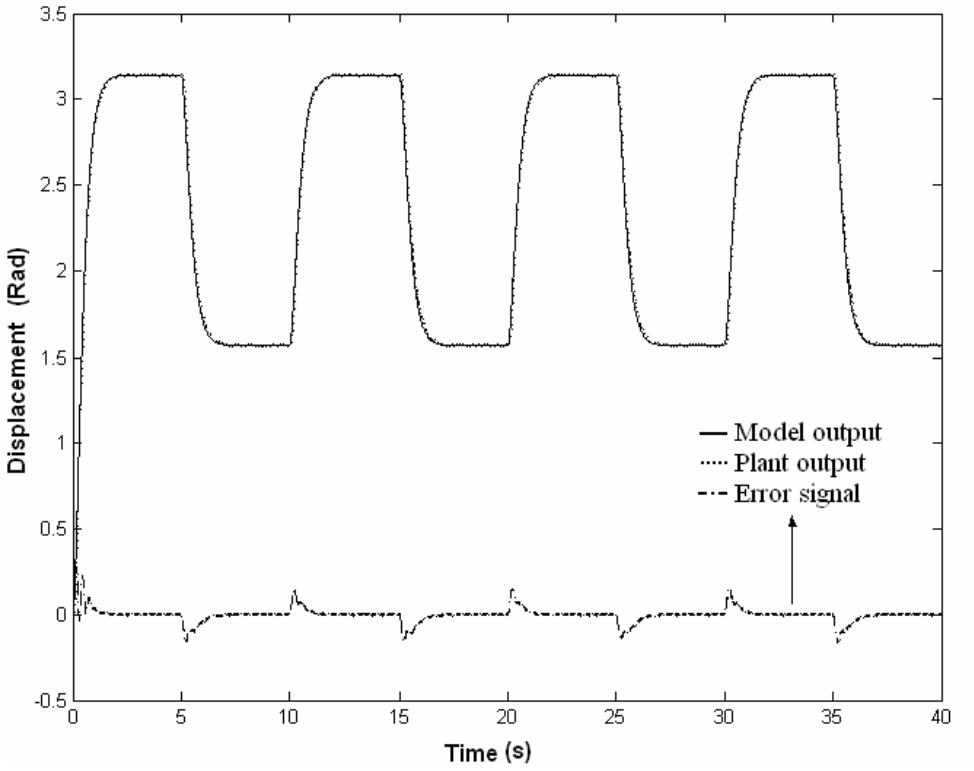

Figure 5.4 Simulation results for $\alpha = 10^{-2}$, $b >> c$, T=1 ms.

ii. $\boldsymbol{P} = \begin{bmatrix} 7.625 & 0.0625 \\ 0.0625 & 0.5078 \end{bmatrix}$ is for $\boldsymbol{Q} = \begin{bmatrix} 2 & 1 \\ 1 & 8 \end{bmatrix}$. Contrary of previous case, this time the coefficient of e_2 is approximately $0.5078/0.0625 \approx 8$ times bigger than e_1. As shown in Fig. 5.5 adaptive tracking is much worse than the previous one. The plant doesn't follow the model properly, especially around $\pi / 2$ rad under steady-state condition. Shortly, the case of ($b << c$) is more terrible than ($b >> c$).

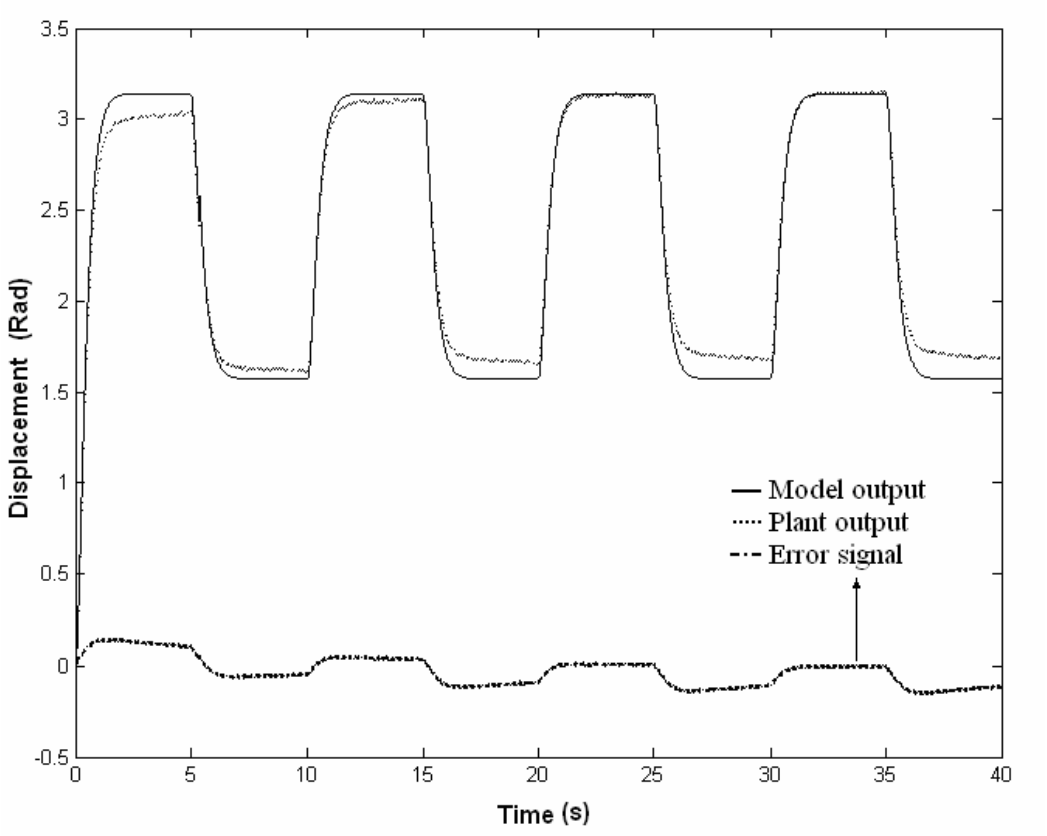

Figure 5.5 Simulation results for $\alpha = 10^{-2}$, $b << c$, T=1 ms.

iii. $\boldsymbol{P} = \begin{bmatrix} 1 & 0 \\ 0 & 0.0625 \end{bmatrix}$ for $\boldsymbol{Q} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

As shown and mentioned in Eqs. 3.23-a and 3.23-b, only e_2 is effective on the adaptive laws. As shown in Fig. 5.6, adaptation can not occur and the plant output goes to away from the model output little by little. In order not to meet such a result it is advise that all the elements of matrix Q are chosen different from zero.

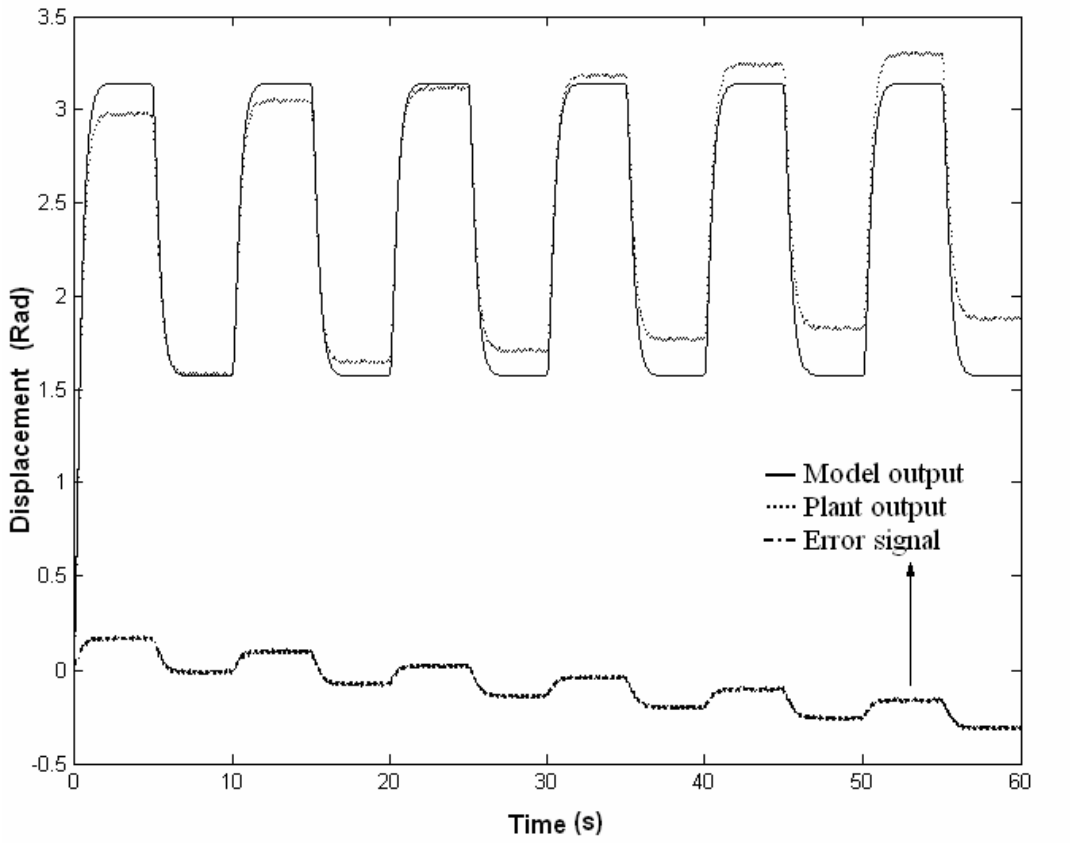

Figure 5.6 Simulation results for $\alpha = 10^{-2}$, b=0, T=1 ms.

5.3 EFFECTS OF SIMULATION STEP SIZE *T* ON THE PERFORMANCE

Now, we investigate how adaptive tracking performance is affected if simulation step size (T) is chosen larger than 1ms. We consider two different T values 10 ms and 100 ms. The simulation results with respect to the these values are shown in Fig. 5.7 and Fig. 5.8 respectively. Although the simulation step size T (10 ms) is smaller than T_p (64 ms) and T_{mi} (250 ms), adaptive tracking performance shown in Fig. 5.7 is less successful than the one in Fig. 5.3 whose T value is (1ms). Hence, even if $T<T_{pi}$ *and* $T<T_{mi}$, the larger simulation step size *T,* the lower performance we get.

Because, simulation step size T (100 ms) is not smaller than T_p (64 ms) adaptive tracking doesn't occur as shown in Fig. 5.8.

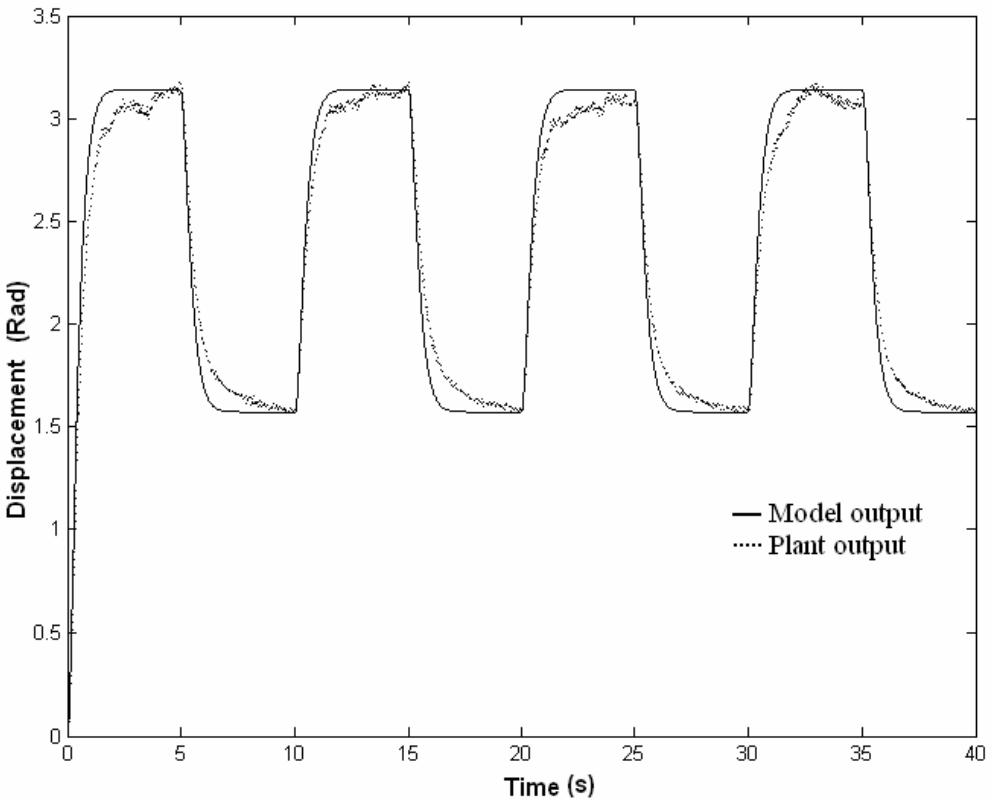

Figure 5.7 Simulation results for $\alpha = 10^{-2}$, $b \cong c$, T=10 ms.

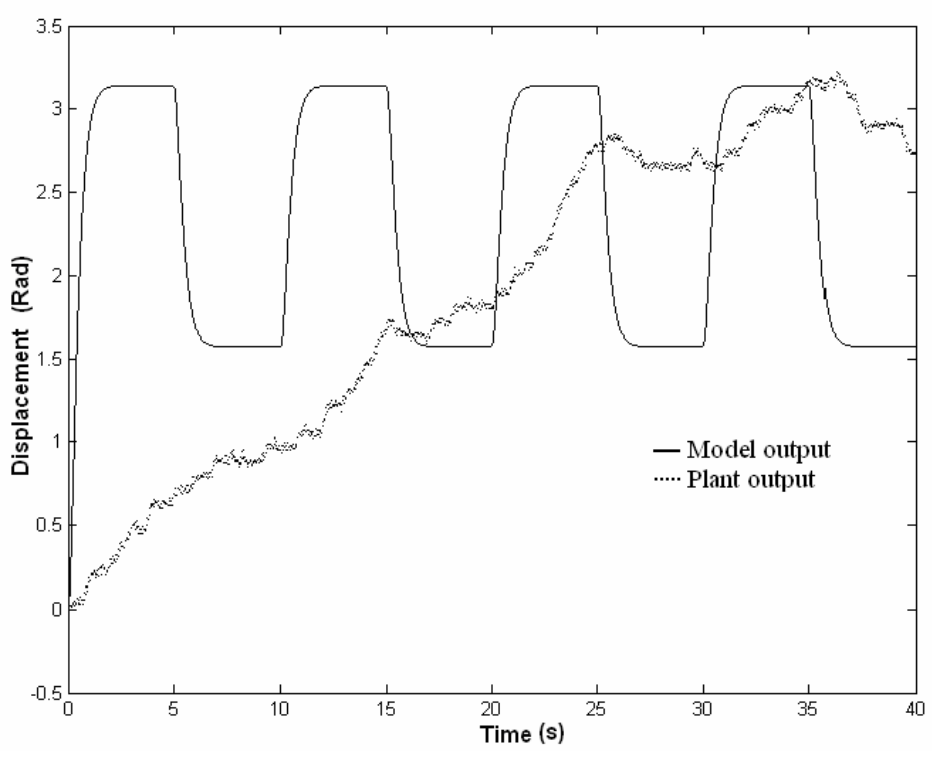

Figure 5.8 Simulation results for $\alpha = 10^{-2}$, $b \cong c$, T=100 ms.

5.4 EFFECTS OF REFERENCE MODEL ON THE PERFORMANCE

Now, we investigate the success of adaptive tracking for a faster and less stable reference model. ζ and w_n are chosen as 0.2 and 8 respectively.

We get the following results by using the formulas 5.1 through 5.5.

Percent maximum overshoot = 52.66,

Quality factor of denominator polynomial = 2.5 > 1,

$t_d \cong 0.01787$ s,

$t_r \cong 0.01615$ s,

$t_s \cong 2$ s for $0 < \zeta < 0.69$.

From Eq. 4.30, $\boldsymbol{A_m}$ and $\boldsymbol{B_m}$ are

$$\boldsymbol{A_m} = \begin{bmatrix} 0 & 1 \\ -64 & -3.2 \end{bmatrix}, \quad \boldsymbol{B_m} = \begin{bmatrix} 0 \\ 64 \end{bmatrix}.$$

Q is chosen as $\boldsymbol{Q} = \begin{bmatrix} 29 & 1 \\ 1 & 1 \end{bmatrix}$.

$\boldsymbol{P}$ is solved from the Lyapunov Equation (Eq. 4.33)

$$\begin{bmatrix} 0 & -64 \\ 1 & -3.2 \end{bmatrix}\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -64 & -3.2 \end{bmatrix} = -\begin{bmatrix} 29 & 1 \\ 1 & 1 \end{bmatrix}$$

of which solution is

$$\boldsymbol{P} = \begin{bmatrix} 14.2563 & 0.2266 \\ 0.2266 & 0.2271 \end{bmatrix}.$$

$\boldsymbol{P}$ satisfies criterions that are mentioned at the end of Section 3.2. Namely, the elements of bottom row be close each other.

From Eqs. 4.31 and 4.32, adaptive laws are

$$\begin{bmatrix} F_1(k+1) \\ \mathrm{F}_2(k+1) \end{bmatrix} = \begin{bmatrix} F_1(k) \\ \mathrm{F}_2(k) \end{bmatrix} - \alpha \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \begin{bmatrix} 0 & 64 \end{bmatrix} \begin{bmatrix} 14.2563 & 0.2266 \\ 0.2266 & 0.2271 \end{bmatrix} \begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix},$$

$$g(k+1) = g(k) + \alpha\, \Theta_r(k) \begin{bmatrix} 0 & 64 \end{bmatrix} \begin{bmatrix} 14.2563 & 0.2266 \\ 0.2266 & 0.2271 \end{bmatrix} \begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix}.$$

As used in Eqs. 3.21a-b, $\boldsymbol{b}_m^T \boldsymbol{P}$ is abbreviated by $\boldsymbol{s}^T = \begin{bmatrix} 14.5 & 14.5313 \end{bmatrix}$.

So adaptive laws are simply written as

$$\begin{bmatrix} F_1(k+1) \\ \mathrm{F}_2(k+1) \end{bmatrix} = \begin{bmatrix} F_1(k) \\ \mathrm{F}_2(k) \end{bmatrix} - \alpha \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \begin{bmatrix} 14.5 & 14.5313 \end{bmatrix} \begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix},$$

$$g(k+1) = g(k) + \alpha\, \Theta_r(k) \begin{bmatrix} 14.5 & 14.5313 \end{bmatrix} \begin{bmatrix} z_1 - \Theta \\ z_2 - w \end{bmatrix}.$$

Adaptation period $\boldsymbol{T}$ is chosen 1ms in this simulation. The time constant of the plant is still $T_p = 1/15.66 = 6.38 \times 10^{-2}\, s \cong 64ms$, and the time constants of the model yield $\mathrm{T_{m1}}$=625 ms for the exponential decaying and $\mathrm{T_{m2}}$=125 ms for the oscillations period. In fact this amount of oscillation period corresponds to a much smaller (approximately 4 times) time constant. Simulation step size T is much smaller than plant's and model's time constants ($T << T_{pi}$ and $T << T_{mi}$). It satisfies criterions that mentioned at the end of Section 3.2.

Reference signal is a square wave, its frequency is chosen as 0.1 Hz and its amplitude is chosen between $\pi/2$ and π in this simulation. Simulation result with respect to the mentioned model is shown Fig. 5.9. Because model trajectory or desired trajectory changes very rapidly due to faster behavior of the model, the plant does not follow the model properly especially around $\pi / 2$ rad.

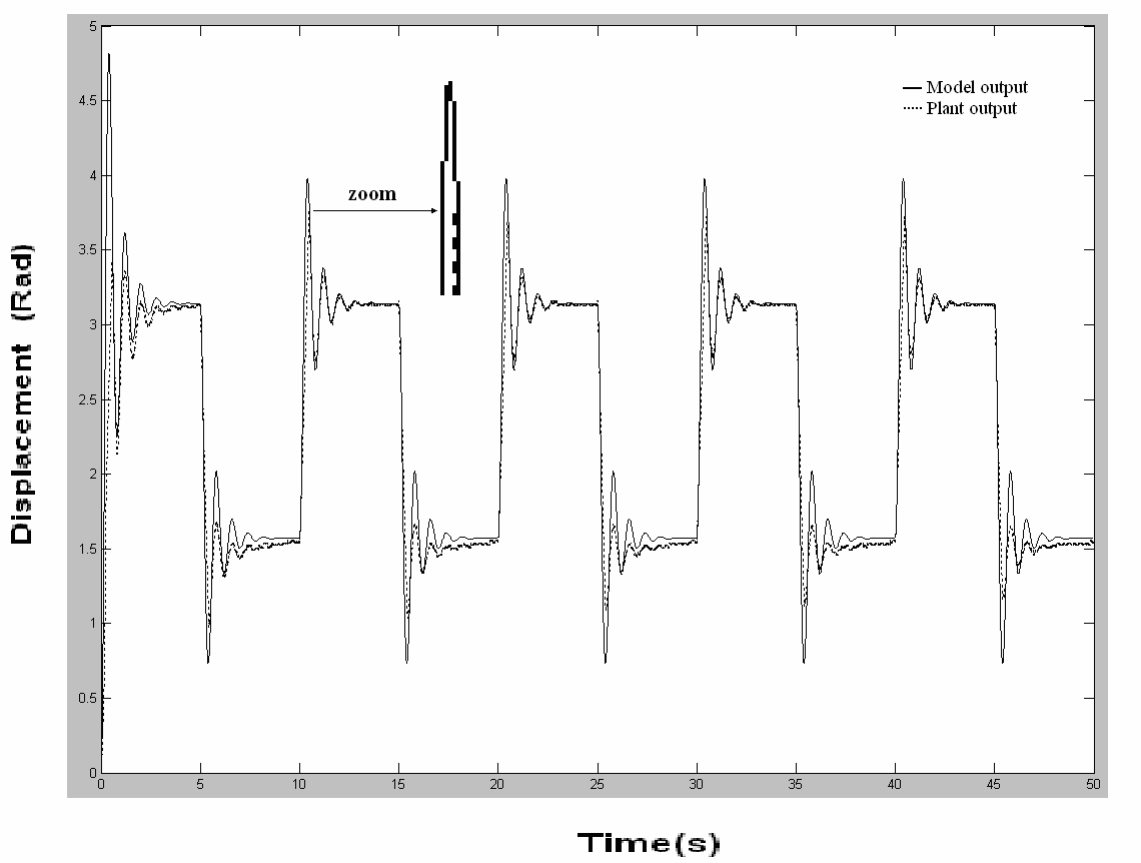

Figure 5.9 Simulation result with respect to faster and less stable model.

5.5 EFFECTS OF DISTURBANCES ON THE PERFORMANCE

We will investigate the success of the adaptation mechanism against to the disturbance effects. We apply disturbance effects with two ways which are changeable armature resistance (R_{ext}) and load torque (T_L).

5.5.1 Disturbances with Changeable Armature Resistance

To begin with, we will apply disturbance effect by means of the disturbance potentiometer (R_{ext}) which is connected in series with the armature resistance R_a of dc motor. The experimental scheme is shown in Fig. 4.13. The reference model and the values of other parameters which are used for this section are same as those used for Fig. 5.3 namely, $\alpha = 10^{-2}$, b = 0.0625, c = 0.0703, T =1 ms.

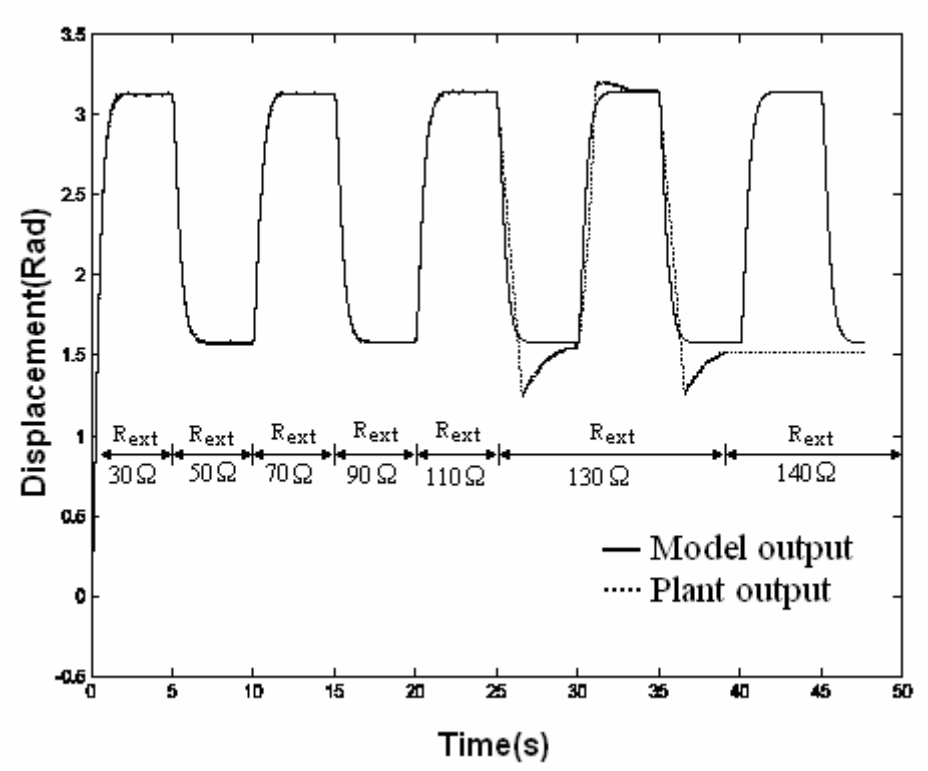

Figure 5.10 Simulation result with respect to armature resistance variations

As shown in Fig. 5.10 the value of disturbance potentiometer (R_{ext}) is increased by gradually within specified time intervals. No appreciable deviation occurs between resistance levels up to $110\,\Omega$, plant follows model very well. At time 25 second, disturbance resistance is increased to the value of 130 Ω and adaptive tracking becomes poor. At time 39 second disturbance resistance is increased to the value of 140 Ω and motor shaft stops. At first glance, this situation may be perceived as weakness of adaptation mechanism. As mentioned in the Section 4.2.2, dc motor has dead zone between -1 and 1 V levels of applied voltages E_a. R_a is already known as 15.36 Ω from Section 4.2.1, under these circumstances, from ohm law, dead zone armature current I_a interval is between -65mA and +65mA. In other words motor torque becomes not sufficient to overcome the static resistance and motor shaft doesn't rotate between these levels. Total armature resistance is $R_T = 140 + 15.36 = 155.36\Omega$ for $R_{ext} = 140\,\Omega$. Then maximum armature current is $I_a = 10 / 155.36 = 64.36$ mA for maximum allowed input voltage $E_a = 10\,V$. This value is inside of dead zone level. Hence, the stopping of the shaft for R_{ext} = 140 Ω is not an abnormal situation. Armature current I_a is 68.79 mA for $R_{ext} = 130\,\Omega$. This value is close to dead zone, for this reason adaptive tracking is poor. Figure 5.10 shows that the value of sudden

change of $20\,\Omega$ in R_{ext} does not spoil the control mechanism since there is not an perceptible difference on the values of the models and plant outputs (See subsequent intervals from 0 to 25 ms). To view the effect of abrupt change of R_{ext} within the region where the adaptation mechanism work well ($0 \le R_{ext} \le 110$), we cary R_{ext} with parallel switch from 0 to $110\,\Omega$ and 110 to $0\,\Omega$ suddenly. Simulation results are shown in Fig. 5.11.

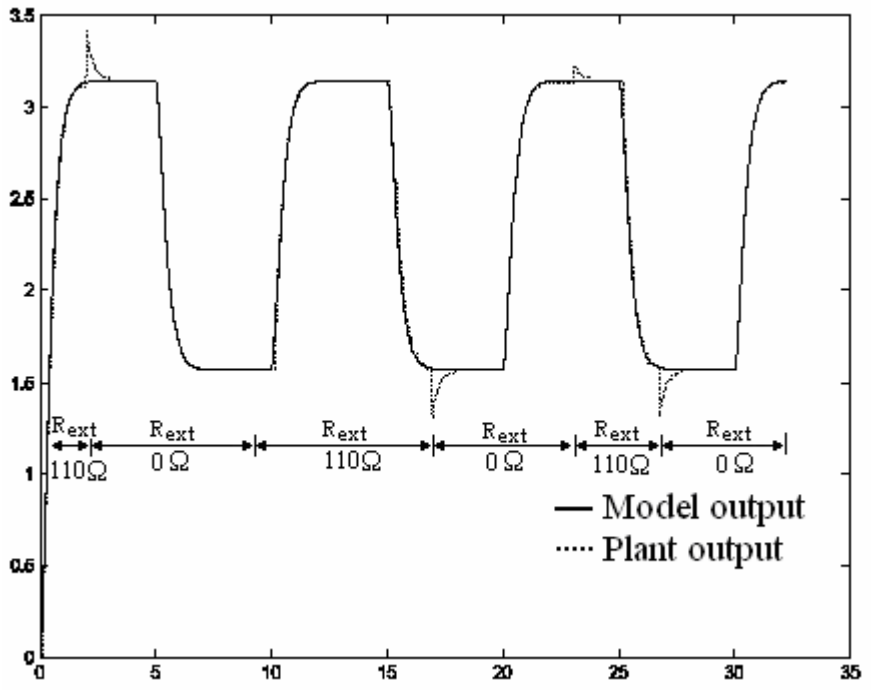

Figure 5.11 Simulation results with respect to abrupt armature resistance variations (0 to $110\,\Omega$ and 110 to $0\,\Omega$)

The switch wich is connected in parallel with disturbance potentiometer (R_{ext}) is on-off within specified time intervals. Although small deviations occur after switching time, adaptation mechanism adjust itself very rapidly in almost 1s.

5.5.2 Disturbances with Changeable Load Torque

Now, we will apply disturbance effect by means of disturbance motor M_D which is identical with M_1. M_D acts as load for M_1 and applies disturbance effect for displacement. Disturbance torque or load of M_1 (T_L) is adjusted by the armature current I_D of M_D. As shown in Figure 5.12, disturbance current I_D is increased by gradually within specified time intervals. No deviations occur between disturbance current levels up to 0.4 A, plant follows

model very well. At time 33 second, disturbance current is increased to the value of 0.5 A and, after 2 seconds plant output leave the model output when the reference input reversed. As mentioned in Section 4.4.1, maximum output current of driver of M_1 is limited with 0.5 A. At time 33 second, the currents of identical motors are equal, therefore their torqueses are same, and then the directions of torques are converse of each other, so motor shaft doesn't rotate.

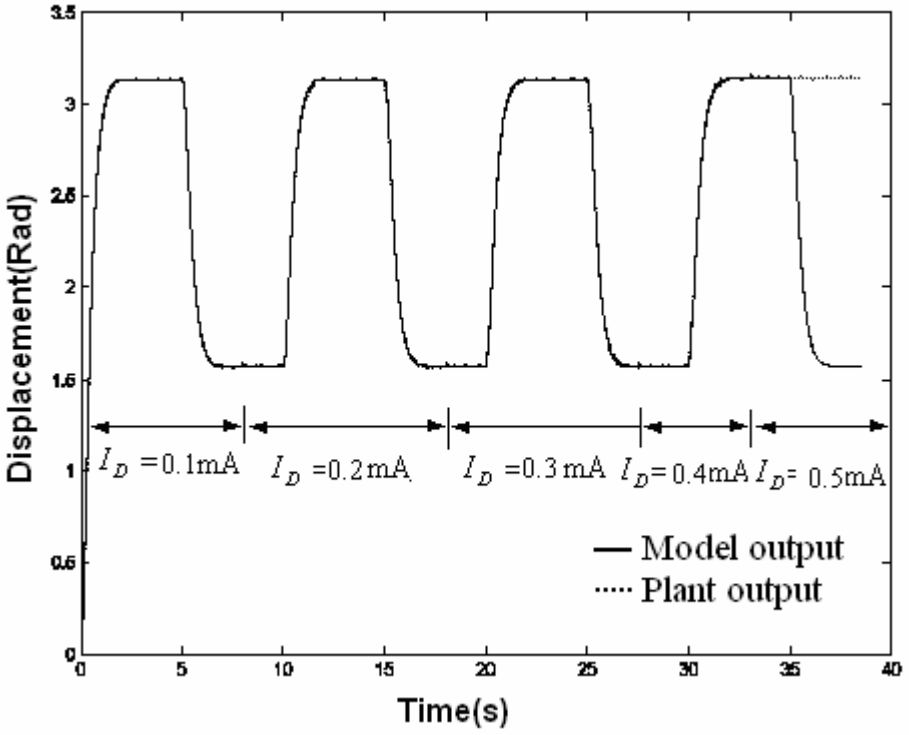

Figure 5.12 Simulation result with respect to torque disturbances.

In reality, the motor may be subject to two types of saturation or limitations. One limitation is that as the armature current increases due to the increase in E_a, the magnetic circuit will saturate, so that the motor torque cannot exceed a certain maximum value. The second limitation is due to the maximum current that the motor can handle due to the heat dissipation rating of the motor. Amplifier gain or motor driver gain is subject to magnetic saturation and heat dissipation. For these reasons, the maximum output voltage and maximum output current of motor driver are limited 10V and 0.5A respectively. Unless these limitations are not to be for motor and its driver, MRAC overcomes almost all excessive load parameter variations and adaptive tracking is continuous.

Figure 5.12 shows that the value of sudden change of 0.1mA in disturbance current of M_D does not spoil the control mechanism since there is not an perceptible difference on the values of the models and plant outputs (See subsequent intervals from 0 to 0.4mA). To view the effect of abrupt change of T_L within the region where the adaptation mechanism work well ($0 \le I_D \le 0.4\,mA$), we cary I_D from 0 to 0.4mA and 0.4 to 0 mA suddenly. Simulation results shown in Fig. 5.13. The switch wich is connected in series with disturbance current source is made on-off within specified time intervals. Although small deviations occur just after the switching time, adaptation mechanism adjust itself very rapidly in almost 1s.

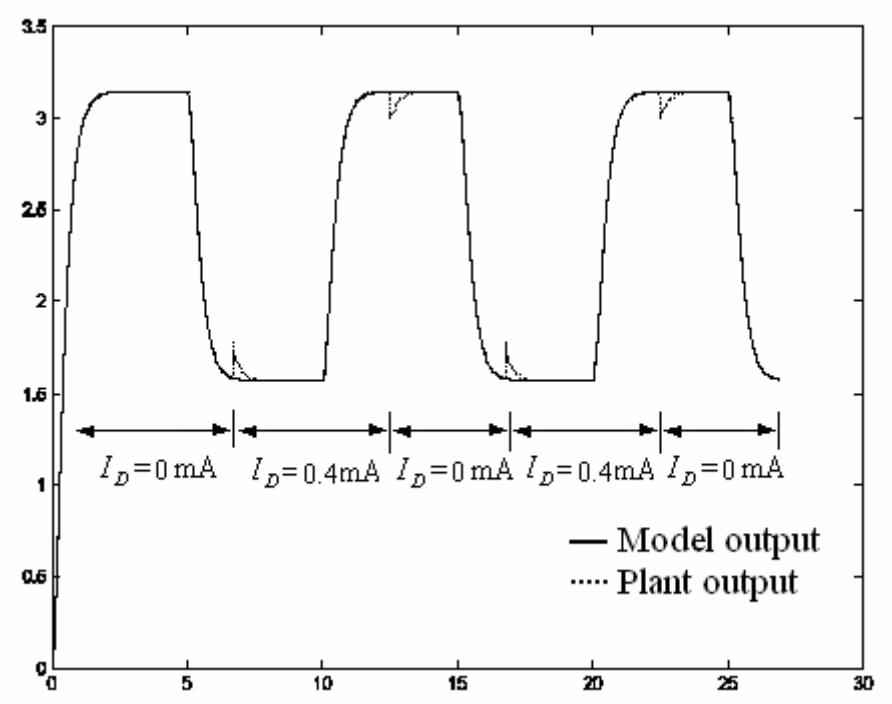

Figure 5.13 Simulation result with respect to abrupt torque disturbances.

CHAPTER 6

CONCLUSIONS

Model Reference adaptive control of a permanent DC motor is investigated in this study. After an introduction of different control problems and MRAC, the general theory about the method and its application to the considered motor (plant) is treated. Identification of motor parameters by experimental methods are evaluated to know the motor for handling the proper choice of the reference model and the correct judgement of the MRAC method. Other than gaining the theoretical and practical knowledge about MRAC, identification problem, matlab tools, this work has also been beneficial in getting acquaintance with the interactive study involving computer and real life.

The main results of this study may be summarized as in the following bullets;

- Model reference adaptive control is one of the various techniques of solving control problem when the parameters of the controlled process are poorly known or vary during normal operation. For this reason, a model reference adaptive control method is proposed to control the position of a dc motor without requiring any fixed motor parameter. Experimental results show how well this method controls the position of the motor.
- This method is only suitable for plants without finite transmission zeros.
- This method adjusts feedback gains of states and feed forward gain of closed loop plant, equalizes them to model's.
- If plant dynamics can be expressed with lower order transfer function, reference model can be chosen as lower order.
- The displacement of motor is measured by encoder. Encoder output has insignificant noise, hence, we don't use filter for plant output.
- To choice of small adaptive gain α may result in slow convergent rate whereas large α may make the differential equations stiff and difficult to solve numerically on the computer.
- Simulation step size T must be bigger than the computation time T_c ($T > T_c$) , otherwise simulation program does not work properly and it gives error message. But the large

simulation step size T reduces the adaptive tracking performance. Computation time T_c involves the evaluation of adaptive laws and it depends on hardware features. It is not measured precisely or properly and already it is not unnecessary, because Simulink gives error message automatically for ($T \leq T_c$).

- Simulation step size T must be smaller than plant's and model's time constants ($T < T_{Pi}$ and $T < T_{mi}$) otherwise adaptation mechanism is insufficient for perception and correction to the parameters variations of plant.
- The absolute values of bottom row elements of ***P*** must be close each other ($b \cong c$) , these elements are coefficients of error signals ***e*** (e_1 and e_2) on the adaptive laws. Otherwise the effect of elements of ***e*** on the adaptive laws are different ratios and adaptive tracking performance decreases. That is adaptation is more efficient when all entries of the error vector are almost equally weighted for adaptation.
- If except diagonal elements, remaining elements of ***P*** are zero (b=0), only e_2 is effective on the adaptive laws. In these circumstances, adaptation doesn't occur. In order not to meet with such a result advise that all the elements of matrix Q which is used in the Lyapunov equation (Eq. 3.16) are chosen different from zero.
- In spite of change of motor parameters during working time, MRAC adjusts plant's input so that output of the plant becomes as model output.
- The motor is subjected to two types of limitations, one is magnetic saturation and second is heat dissipation. For these reasons, the maximum output voltage and maximum output current of motor driver are limited within specific intervals. Unless these limitations are not exceeded for motor and its driver, MRAC overcomes all excessive load parameter variations and adaptive tracking is continuous.
- The parameter variations with time which are faster as compared to simulation step size ***T*** are not perceived and corrected by MRAC such as white noise and high frequency disturbances.
- However, the method might be improved to apply the plants with finite transmission zeros and this is the subject of the future work.

APPENDIX A

LYAPUNOV STABILITY

Consider the nonlinear, time-varying system (Butler, 1992):

$$\dot{x} = f(x,t). \qquad \text{(A.1)}$$

Here, f is an n x 1 vector function and x is the n x 1 state vector. An equilibrium point x^* of system (A.1) is characterized by $f(x^*) = 0$. If the state x of system (A.1) is situated at an equilibrium point at $t = 0$, it will stay on the equilibrium point for all $t > 0$. If the initial state isn't exactly equal to the equilibrium state ($\|x(0)\| = \|x^*\| + \delta$), there are four possibilities.

1. The state x can be made to remain in any specified vicinity ε of x^* by choosing a sufficiently small deviation δ of x^* at $t = 0$. However, x isn't guaranteed to converge to x^*. This behavior is called stable.

2. The system isn't stable. It is than said to be unstable. Note that if a system is unstable, this doesn't imply divergence from x^*. It merely states that it is possible to specify an ε which doesn't allow any δ. A well known example of this is a harmonic oscillator which is oscillating with a specific frequency and amplitude: $x(t) = A * \sin(wt)$. While x always lies between -*A* and +*A*, specifying $\varepsilon > A$ allows any initial deviation $\delta < \varepsilon$. However, no value of δ can be found that allows an $\varepsilon < A$ because x will always exceed ε for some time t.

3. The state x will converge to x^* as long as the initial deviation δ of x^* is smaller than some boundary *R*. More formally, if $\delta < R$, for an arbitrarily close vicinity μ of x^* a time t^* exists such that for all $t > t^*$, $\|x - x^*\| < \mu$. This notion is called asymptotic stability. Note that in this definition, starting within a boundary *R* from x^* doesn't imply that x will remain within *R* for all $t > 0$, but only that $\|x - x^*\| \to 0 \; as \; t \to \infty$.

4. If the asymptotic stability is guaranteed for any initial deviation δ, the asymptotic property is said to be global. To check to which stability class a given system of the form

(A.1) belongs, Lyapunov's direct method can be applied. Lyapunov's method can only investigate stability properties of an equilibrium point $x^* = 0$. Hence, other equilibrium points have to be transformed to $x^* = 0$ by a transformation $x' = x - x^*$. In applying Lyapunov's method, first a Lyapunov function $V(x)$ is defined. This function can be considered as a sort of energy function, while it has similar properties to the energy stored in the system. The Lyapunov function itself must satisfy

$V(x)>0, \quad \forall x \neq 0$.

In addition to be comparable to energy function $V(x)$ should be monotonically increasing, and go to infinity as $\|x\| \to \infty$;

$0 < \alpha\|x\| < V(x); \quad \alpha > 0$

$V(x) \to \infty \quad as \quad \|x\| \to \infty$.

Now, it can be felt that if the stored energy in a system decreases as time passes, all energy will eventually leave the system and equilibrium $x = 0$ will be reached. Similarly, if the time derivative of the Lyapunov function is always negative and hence V is decreasing with time, V will eventually become zero because V is monotonous. As V=0 implies x=0, also due to the monotonous character of V, a negative definite $\dot{V}$ guarantees asymptotic stability;

$\dot{V} < 0, \quad \forall x \neq 0$.

To calculate $\dot{V}$, partial derivatives of V with respect to the elements of x are needed;

$$\dot{V}(x,t) = \frac{dx}{dt} \cdot \frac{\partial V}{\partial x} = f(x) \cdot \frac{\partial V}{\partial x}.$$

Hence, the partial derivative $\frac{\partial V}{\partial x}$ must be continuous. The above requirements on V guarantee global asymptotic stability. By using less strict requirements, other forms of stability are obtained. For example, if $\dot{V}$ is negative semi-definite, which implies that $\dot{V} = 0$ for some $x \neq 0$, the stability is no longer asymptotic. If V isn't monotonously increasing with $\|x\| \to \infty$, the stability isn't global. Note that Lyapunov's method provides a stability guarantee: if the requirements mentioned are met, the system is guaranteed to be globally asymptotically stable. However, if the requirements aren't satisfied, the system may still be

stable. In addition, the choice of the Lyapunov function V is crucial in the stability check. Different Lyapunov functions may give different stability results.

For a linear system $\dot{x} = Ax$, let us consider a quadratic Lyapunov function: $V = x^T P x$, in which P is a symmetric positive definite matrix. Then;

$$\dot{V} = \dot{x} P x + x^T P \dot{x} = -x^T Q x$$

in which

$$-Q = A^T P + PA .$$

According to Lyapunov's theorem, a positive definite symmetric matrix Q always yields a positive definite symmetric matrix P if the system $\dot{x} = Ax$ is asymptotically stable.

APPENDIX B

THE FEATURES OF APPLIANCES

B.1 MOTOR SPECIFICATIONS

- M1 and M_D are identical permanent magnet dc motor.
- Maximum armature current is 0.5 A
- Maximum armature voltage is 10 V
- Nominal speed is 5700 RPM.
- Armature resistance is 15.36 Ω
- Armature inductance is 0.42 mH

B.2 ENCODER SPECIFICATIONS

- Typical dual-channel encoder
- It produces 100 square waves per rotation.

B.3 DATA ACQUISITION BOARD SPECIFICATIONS

The MF 614 multifunction I/O card is designed for the need of connecting PC compatible computers to real world signals. The MF 614 contains a 100 kHz throughput 12 bit A/D converter with sample/hold circuit, four software selectable input ranges and 8 channel input multiplexer, 4 independent 12 bit D/A converters, 8 bit digital input port and 8 bit digital output port, 4 quadrature encoder inputs with single-ended or differential interface and 5 timers/counters. The card is designed for standard data acquisition and control applications and optimized for use with Real Time Toolbox for MATLAB®.

B.4 HARDWARE SPECIFICATIONS

- Microprocessor is Pentium 4, 3 GHz
- RAM is 512MB
- Hard disk is 20GB

B.5 SOFTWARE SPECIFICATIONS

- Matlab version is 7.1
- Simulink version is 6.3
- System Identification Toolbox version is 6.1.2

APPENDIX C

THE MATLAB PROGRAM CODES

C.1 SIMULINK SCHEME OF MRAC OF A DC MOTOR DISPLACEMENT

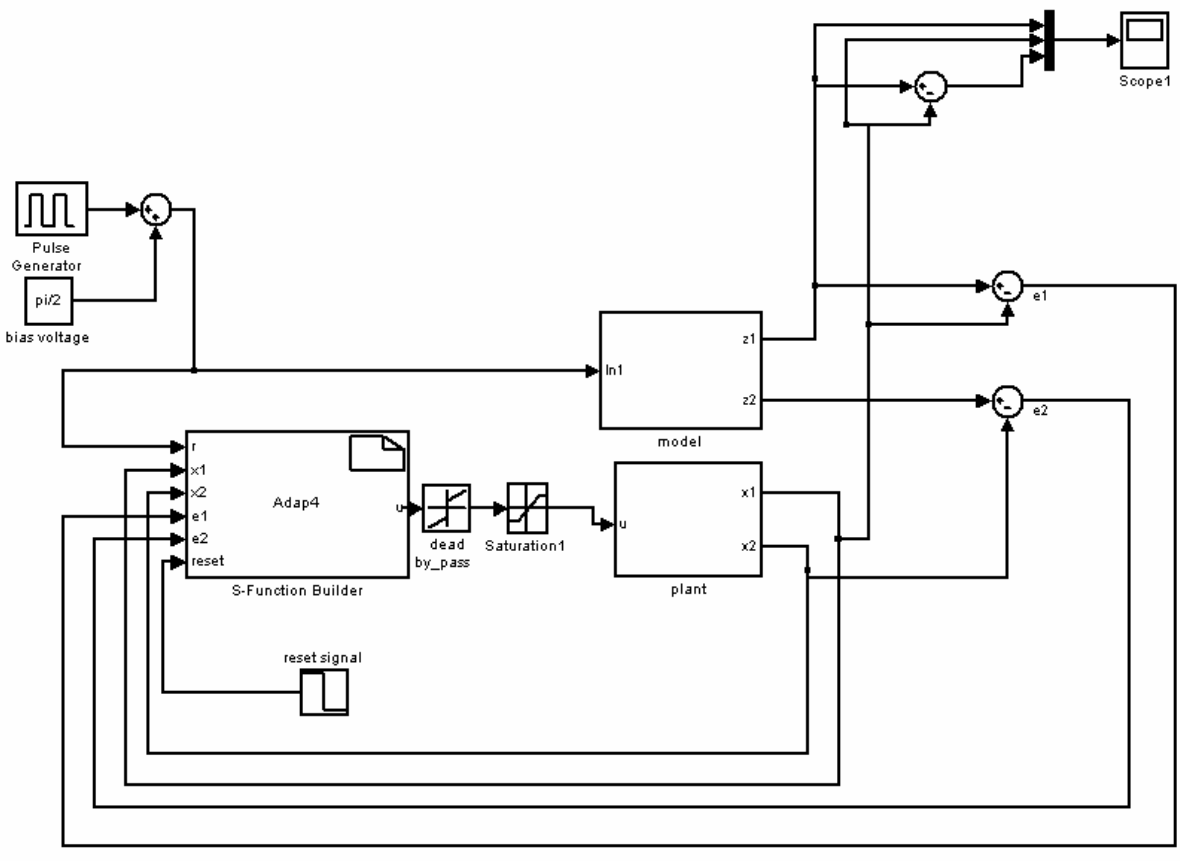

Figure C.1 Simulink scheme of MRAC

C.1.1 Inside of Model Block

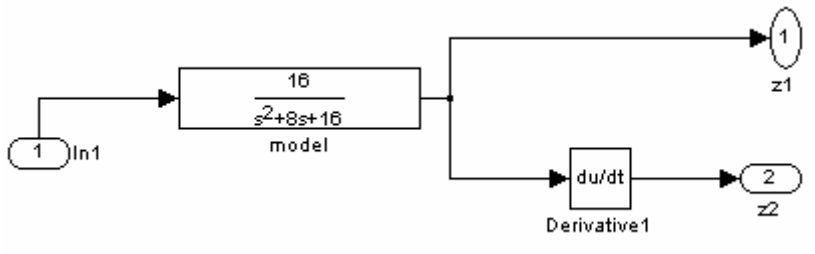

Figure C.2 Interior of Model Block

C.1.2 Inside of Plant Block

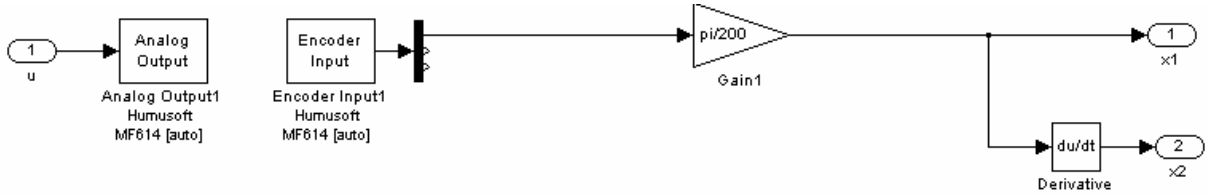

Figure C.3 Interior of Plant Block.

C.1.3 Inside of S-Function Builder Block (Adaptation Mechanism)

```
%this code is written with C++ program
 Static double F1=0.0, F2=0.0, g=0.0, a=.01, s1=1, s2= 1.1250;
  If (reset [0]>0.5)
  F1=F2=g=0.0;
  F1 -= a*(s1*x1[0]*e1 [0] + s2*x1[0]*e2 [0]);
  F2 -= a*(s1*x2[0]*e1 [0] + s2*x2[0]*e2 [0]);
  g += a*r [0]*(s1*e1 [0] +s2*e2 [0]);
  U [0] = r [0]*g - F1*x1[0] - F2*x2[0];
```

C.2 ORDER REDUCTION of MOTOR TRANSFER FUNCTION with MATLAB

```
%ORDER REDUCTİON WITH MATLAB'S BALREAL AND MODRED COMMANDS
TFIDE3=TF ([4.827E7], [1 3.656E4 5.729E5 0])
[NUM, DEN] = TFDATA (TFIDE3,'V');
SYS=TF([NUM(1,4)],[DEN(1,1) DEN(1,2) DEN(1,3)]);
[SYSB, G] = BALREAL (SYS);
SYSR = MODRED (SYSB, [2],'DEL');
SYSR=TF (SYSR);
[NUM, DEN] = (TFDATA (SYSR,'V'));
FPRINTF ('ORDER REDUCTION WITH MATLAB' S BALREAL COMMAND =%5.15E\N')
TFBAL=ZPK(TF([NUM(1,2)],[DEN(1,1) DEN(1,2) 0]))%END OF PROGRAM
```

REFERENCES

Aiko Miyamura , Hidenori Kimura, April 2002, "Stability of feedback error learning scheme", *Systems & Control Letters, Automatica, Volume 45, No.4, Pages 303-316*.

Aström, K. J. and Wittenmark,B, *Adaptive Control*, Addison Wesley, New York, 1995.

Butler, H., *Model Reference Adaptive Control*, Prentice Hall, New York, 1992.

Chak C. K., Gang Feng and T. Hesketh, March 1997, "Multirate adaptive optimal control with application to dc motor" *computers elect. engng* vol. 23, no. 2, pp. 65-79.

Chaoa C.L., J. Neoub, October 2000, "Model reference adaptive control of air-lubricated capstan drive for precision positioning",*Precision Engineering*, Vol. *24*, *Issue 4* , pp. *285–290.*

Chien Chiang-Ju and Chia-Yu Yao , May 2004, "Iterative learning of model reference adaptive controller for uncertain nonlinear systems with only output measurement*", Automatica, Volume 40, Issue 5, Pages 855-864.*

Costa Ramon R., Liu Hsu , Alvaro K. Imai , Petar Kokotovic, July 2003, "Lyapunov based adaptive control of MIMO systems", *Automatica, Volume 39, Issue 7, Pages 1251-1257.*

Ioannou, P. A., *Robust Adaptive Control*, Prentice Hall, New York, 1994.

Karadeniz, M. et al, 2004 "*Adaptive Pole Placement Control of a DC Motor Speed*", J. Fac.Eng. Arch. Gazi Univ. Vol 19, No 3, 327-334.

Krstic Miroslav and Andrzej Banaszuk, June 2005, "Multivariable adaptive control of instabilities arising in jet engines", *Control Engineering Practice, In Press, Corrected Proof, Available online 13 June 2005*

Kojabadi H. M., 2005, "Simulation and experimental studies of model reference adaptive system for sensorless induction motor drive", *Simulation Modelling Practice and Theory, Volume 13, Pages 451-464*

Kuo, B. C., *Automatic Control Systems*, Prentice Hall, New Jersey, 1995.

Lee T. H, S. S. Ge, C. P. Wong , December 1998, "Adaptive neural network feedback control of a passive line of sight stabilization system", *Mechatronics, Volume 8, Issue 8, Pages 887-903*.

Lee T.H., K.K. Tan, S.N. Huang, H.F. Dou, December 2000, "Intelligent control of precision linear actuators", *Engineering Applications of Artificial Intelligence, Volume 13, Issue 6, Pages 671-684.*

Makoudi M., L. Radouane, July 2000 , "Robust decentralized adaptive control for non-minimum phase systems with unknown or time varying delay", *Automatica*, Vol. *35*, No. 2, pp. *1417-1426.*

Makoudi M. and L. Radouane, August 1999, "A robust decentralized model reference adaptive control for non-minimum-phase interconnected systems"*, Automatica, Volume 36, Issue 7, Pages 1057-1065.*

Marino R, S. Peresada and P. Tomei, May 1998, "Adaptive Output Feedback Control of Current-Fed Induction Motors with Uncertain Rotor Resistance and Load Torque", *Automatica*, Volume 34, Issue 5, Pages 617-624.

McLain Richard B., Michael A. Henson, April 2000, "Principal component analysis for nonlinear model reference adaptive control", *Computers and Chemical Engineering*, Volume 24, Pages 99-110.

Mirkin Boris M., Olof Gutman, October 2005, "Output feedback model reference adaptive control for multi-input–multi-output plants with state delay", *Systems & Control Letters, Vol. 54, Issue 10, pp. 961–972.*

Park C., Kwon H., 2004, "Simple and robust speed sensorless vector control of induction motor", *Electric Power Systems Research, Volume 71, Pages 257-266.*

Sinha P.K., A.N. Pechev, August 1999, "Model reference adaptive control of a maglev system with stable maximum descent criterion", *Automatica*, Vol. *35*, No. 8, pp. *1457-1465.*

Taware A., Gang Taoa, Nilesh Pradhana, Carole Teolisb, March 2003, "Friction compensation for a sandwich dynamic system"*, Automatica*, Volume 39, Issue 3, Pages 481-488.

Tian Z., K.A. Hoo, November 2003, "Transition control using a state shared model approach", *Computers and Chemical Engineering, Vol. 27, Issue 11, pp. 1641-1656.*

Tsai Y . K . and Y . H . Lin, July 1997," Adaptive modal vibratıon control of a fluid-conveying cantilever pipe" *Journal of Fluids and Structures,Volume 11, Issue 5, Pages 535-547.*

Wang Hong, F Zhen J. Huangs And Steve Daley, February 1997, "On the Use of Adaptive Updating Rules for Actuator and Sensor Fault Diagnosis", *Automatica, Volume 33, Issue 2,, Pages 217-225*

Yasser Eldeeb and W H. Elmaraghy, July 1998, "Optimal control of a single link manipulator including motor dynamics", *Computer Integrated Manufacturing Systems, Volume 11, Issue 3, Pages 199-205*

Zeng Y., A.D. Araujo and S.N. Singh, 1999 "Output feedback variable structure adaptive control of a flexible spacecraft**",** *Acta Astronautica* Vol. 44, No. 1, pp. 11±22,.

Zhou J.and Y. Wang, October 2005, "Real-time nonlinear adaptive backstepping speed control for a PM synchronous motor, *Control Engineering Practice*, *Volume 13, Issue 10, pp 1259-1269*

Zhong Y. S., September 2005, "Globally stable adaptive system design for minimum phase SISO plants with input saturation", *Automatica*, Volume 41, Issue 9, Pages 1539-1547.