NONLINEAR MODELING AND CONTROL OF A SPARK IGNITION ENGINE IDLE SPEED

HAZEM MOHAMED, B.Sc.

A THESIS SUBMITTED TO THE FACULTY OF ENGINEERING, UNIVERSITY OF MALAYA, IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING SCIENCE.

OCTOBER 1998

ACKNOWLEDGEMENT

I am deeply indebted to my Supervisor, Professor Dr. Mohamed Zaki Abdulmuin, for his encouragement and guidance throughout the study period. Also thanks to Prof. Madya Dr. Masjuki Haji Hasan for co-supervising the research. I am also grateful to all those in the academic, technical and general staff who had in one way or another contributed to the completion of the thesis. Deep thanks to the library staff namely Puan Munisa Yosuf, Fatima, Wahab and Subramaniam. The help of Juneni (Computer lab), Haji Yakop and Dehis is highly appreciated.

Deep thanks to Prof. Dr. Khalid Mohamed Nor for different support, for his encouragement throughout the study period.

Thanks to Dr. Hakim Abas from ITM for his advises and technical help during the run of the experiments in ITM. Special thanks to Prof. Abdul Ghani Ujang for granting permission to use the experimental test facilities of his department.

A lot of thanks should be also paid to my friends, Sayoud Hameida, Husain Ben Younis, Redha, Shafi and Nazrul from electrical department. Thanks to Dr. Khalid melgiht, Abdesselam hamaloui, Hafez and to all my friends.

I hereby declare that this thesis is based on the results found by myself. Materials of the work found by other researchers are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Date Signature

ABSTRACT

The main objective of this research work is to model and control the idle mode of a four cylinder spark ignition engine. The modeling approach is to handle the engine as a multi input multi output nonlinear black box model, where the inputs are the throttle angle and the spark timing and the outputs are the speed and manifold pressure.
The input output data was collected from an experimental four cylinder spark ignition engine, then this data is fitted to a NARMAX model using a radial basis function network trained by the orthogonal least squares algorithm. After that the developed model is validated using nonlinear correlation tests and it could efficiently reproduce the plant dynamics.

The control technique used is the fuzzy control. The fuzzy controller is formulated as a radial basis function network trained by the orthogonal least squares algorithm to estimate the controller parameters. The training data of the network is the entries of an optimal control table derived by analyzing the evolution of the state trajectories of the system under different initial conditions using the cell to cell mapping technique. The developed controller can efficiently stabilize the plant and compensate for different uncertainties and nonlinearities of the plant.

TABLE OF CONTENTS

List of Symbols

Neural Network

x_i	Network input at time instant *i*
κ	Basis or activation function
β_k	Scale or direction parameter
γ_k	Translation or position parameter
α_c	Coordinate parameter
ϕ	Whole mapping of the network
$\hat{\theta}$	Estimated parameter set
$\hat{\theta}_*(m)$	Optimal parameter set
α	Step length
R_i^{-1}	Search direction
$\nabla\hat{f}_i$	Gradient of the cost function
$\overline{V}(q)$	Variance
$E\{\}$	Expectation
N	Number of data points

System Identification

u(t)	Input signal to the plant

$y(t)$	Output signal of the plant
g	Nonlinear function
$v(t)$	Model residuals
φ	Regressor vector

Engine Dynamic Model

u_1	First input
u_2	Second input
y_1	First output
y_2	Second output
$y_{p1}(k+1)$	First output of the model
$y_{p2}(k+1)$	Second output of the model
P_m	Manifold pressure
N	Engine speed [r.p.m]
Th	Throttle angle signal
D	Ignition timing signal
$\hat{y}$	Model estimated output
ε	Model residuals

Radial Basis Function Network and Model

$\| \|$	Euclidean norm
n_h	Number of hidden layer

α_o	Constant term of the expansion
φ	Regressor vector
σ	Spread factor of the Gaussian function
$M\ (=N)$	Initial number of centers
m	Number of the plant outputs
N	Length of the data record
G	Matrix of the basis functions
Y	Output matrix
Θ	Parameter matrix
E	Error matrix
W	Auxiliary regressor matrix (with dimension N*M)
B	Triangular matrix
Γ	Auxiliary parameters matrix
$\hat{\Gamma}$	OLS solution
M_s	Chosen centers
err	Error reduction ratio
ζ	Error reduction ratio criteria
x_p	State vector
$\hat{\theta}$	Outer layer parameters
C_i	Set of chosen centers
A,B,a,b,c,d	Normalizing factors

Model validation

ε	Residuals
μ_{ε}	Mean value of the residuals
$\phi_{u\varepsilon}$	Correlation function between the input and residuals
$\phi_{\varepsilon\varepsilon}$	Correlation function between residuals
H_o	Null hypothesis
H_1	Alternative hypothesis
τ	Time samples
E	Expectation
$\varphi_{\zeta\eta}$	Correlation measure between residuals and the output
$\varphi_{\vartheta\eta}$	Correlation measure between the input, output and the residuals

Controller

x	State vector
Z_i	Integer
h_i	Interval size
e_i	Unit vector in the Z_i direction
z	Vector cell
N	State vector dimesion
$z(n)$	The cell mapping representing the system dynamics
$u(n)$	Control vector

C	Mapping rule
χ	Input universe of discourse
R^n	Input space
V	Output universe of discourse
R^m	Output space
f_i^l, G_j^l	Fuzzy sets
μ_A, μ_B	Membership functions
$\mu_{A\to B}$	Fuzzy implication
x_c	Center of the membership function
$p_j(x)$	Fuzzy basis function
$\overset{Ms}{j}$	Outer layer parameters
C^T	Chosen centers (under load)
c	Chosen centers (no load)
P	Parameters matrix of the outer layer (under load)
p	Parameters matrix of the outer layer (no load)

Acronyms

ANN	Artificial neural network
ARX	Autoregressive model with exogenous input
ARMAX	Autoregressive moving average with exogenous input
BJ	Box-Jenkins model
FIR	Finite impulse response model

ARX	Nonlinear autoregressive model with exogenous input
ARMAX	Nonlinear autoregressive moving average with exogenous input
BJ	Nonlinear Box-Jenkins model
FIR	Nonlinear finite impulse response model
NOE	Nonlinear output error mode
ART2	Adaptive resonance theory
RBF	Radial basis function
CCM	Cell to cell mapping
OCT	Optimal control table

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Engine Control

Microprocessor based control systems have been introduced to the realm of spark ignition engine control by the middle of the 1970's and have become dominating in the 1990's. Instead of mechanical, pneumatic and hydraulic components used by conventional control systems, an electronic control system consists of sensors (pressure, temperature,...), electric actuators (solenoids, dc motors,...) and an electronic control unit (a microprocessor and its attachments). Sensors measure the states of the plant and feed them back to the electronic control unit where the control decision is made and delivered to the engine through the actuators.

Due to increasing performance requirements such as emission control, fuel consumption, driveability, ...etc, the conventional control system are no longer able to meet these requirements. The superior alternative to the conventional engine control system is the microprocessor based control system, and this is due to its superior flexibility, accuracy, tolerance and long term calibration stability.

In general, the aim of an engine control system is to regulate the amounts of air, fuel and spark timing to meet performance requirements under different operating conditions of the engine. The major states to be measured are the speed, manifold pressure, coolant temperature, oxygen content in the exhaust

gases, throttle and exhaust gas recirculation valves positions. The major control signals are fuel control signal, ignition timing signal, idle air signal, and exhaust gas recirculation signal.

Spark ignition engines have different operating modes depending on the operating conditions under which the engine is running (warming, idle, accelerating, decelerating...). Among these modes the one of interest to this research work is the idle mode which is the most frequently encountered operating condition in city driving.

In its idle mode the engine is running on its minimum speed and suffering disturbances coming from different engine accessories such as air condition, automatic transmission (from neutral to drive condition), power steering, and other secondary loads.

The idle mode control problem is a challenging one especially for small size engines as the engine in this mode is running at its minimum torque reserve and significant engine parameter changes can be expected over environmental conditions under which the engine operates.

The key factors to be considered in idle mode control (Cook,1996) include:

a- Engine speed set point: The engine idle speed is kept at its minimum that can guarantee minimum fuel consumption, better combustion quality, enough power for accessories drive requirements, noise, vibration, and harshness properties.Accessory load disturbances: The characteristics and range of operation of these loads determine the complexity of the control design and achievable performance.

b- Control authority and actuator limitations: The actuators involved in the process are subject to constraints imposed by the hardware itself as well as other engine control design considerations.

c- Available measurements: Speed alone or speed and manifold pressure, in addition to accessory load are the states to measure for control.

d- Variations in engine characteristics over the engine operating range: The control system has to be robust to stand for plant parameter changes due to aging and unit-to-unit variability.

1.2 Motivation for Nonlinear Modeling and Control of the Idle Mode

Dynamical systems are always nonlinear in nature. Linearization of dynamic system models is only valid in small range of operation of the plant, but when the range of operation is large the assumption of linearity is no longer valid and the designed linear controller performs poorly. Spark ignition engines are strongly nonlinear plants and their physical dynamical models always contain uncertainties and spurious modeling assumptions and a lot of ad hoc in the experimental parameter determination.

In the idle mode the system is highly nonlinear with time delays which may be of variable length making analytical treatment of the problem very difficult. Time wasted in making assumptions and trying to prove it, in addition to equipment settings, unavailability and other financial and administrative problems are other serious difficulties. In cases like this it could be reasonable to handle the problem as a black box modeling one guided by some a priori

knowledge about the physics of the plant. Simple models may merely portray the input output relationship of engine performance on a black box basis without characterizing the interrelationships among the internal engine processes or components. For control purposes the model need only portray the dominant dynamics while keeping the interrelationships between measured variables consistent with the physical processes (Coat and Fruechte,1983). The black box solution will save a lot of time, cost and the developed model can be used efficiently for control purposes.

Linear controllers are designed on the basis of a linearized model around an operating point and explicitly or partly ignore the delay elements of the model especially for the conventional PID controllers. A linear controller poorly compensates for nonlinearities and model uncertainties may happen in the system while a nonlinear controller can compensate well. A nonlinear controller can compensate for nonlinearities and uncertainties in addition it may be simpler and intuitive in its design (Slotin,1991). As pointed out before the engine as a plant has a strong nonlinear behavior and uncertainties attend in their parametric and nonparametric forms. Linear controllers designed on the basis of inaccurate or obsolete parameters of the plant yield unsatisfactory and unstable performance in addition to their costly and lengthy periods of development.

Nonlinear control techniques if applied can yield a lower in cost, robust and satisfactory performance control system. A common unconventional nonlinear controller is the fuzzy controller, which can be successfully applied to control nonlinear systems. A fuzzy control system, in addition to its ability to handle nonlinearities can be robust enough to compensate for uncertainties which results from abrupt changes of the plant parameters. These capabilities of the

fuzzy controllers make them an interesting control choice for plants which are not enough amenable to conventional control schemes.

1.3 Previous Work in Modeling and Control of the Idle Mode

Before 1970's engine control systems were model free based on calibration of production engines. The area of engine modeling has started to achieve advances since 1970, especially in conjunction with the introduction of microprocessor technology to automotive industry. Most of the first works in idle mode modeling and control rely on considering the overall system dynamics which results in a model of high order (usually near to eighteen state variables). The second step in modeling is to simplify the model order to third or less order (usually second order). The model is then linearized and several modern and conventional linear control techniques are applied to control the engine.

The first work in engine modeling was the work done by Hazell and Flower (1971). Their work resulted in a discrete model based on crank angle events, in addition they performed frequency, z domain and stability analysis. A model developed by Prabhakar et al (1975) was the first one to contain spark advance; throttle and fuel control variables and empirically based approximations of the engine emissions in addition to transmission/drive line.

A tenth order nonlinear model with wide speed and load operating range was developed by Powell (1979). This model contained description for the induction process, engine power system, throttle mechanism, sonic EGR valve and the fuel injection system.

The most comprehensive model is the one developed by Delosh (1981), where all control mechanisms and sensors as well as versions of models suitable for four, six and eight cylinder engines with various derivations.

Dobner (1983) developed a nonlinear engine model for control analysis. In this model the inputs are the throttle angle, the air to fuel ratio, the spark advance, the EGR command, and load torque. The outputs are manifold pressure, exhaust air/fuel, brake torque, and engine speed. For a survey on the history of engine modeling see Powell (1987).

Of the first works done to the idle mode modeling and control, are those developed by Morris and Powell (1983) and Morris et al (1982). In their work they developed a detailed physical model of the engine idle mode consisting of eighteen states variables and then reduced the model order to a five states model.

A two states linear black box model based on frequency response identification and combustion dynamics approximation was developed by Takahashi et al (1985), with a controller designed by the linear quadratic and integral optimum control theory (LQI technique).

A recent physical model and controller design for an eight cylinder port fuel injected engine was developed by Livshiz et al (1994) where the problem was handled as a two inputs two outputs model and the controller is an optimal controller type. Another recent work is the multi inputs-single output model by Cook (1996) with a modern optimized PID controller.

Many linear control techniques based on linear models have been applied to the idle mode control problem. As an example of the conventional technique is the PID controller. Examples of modern techniques are the linear quadratic

control (Powell,(1986), adaptive control (Liubakka et al, 1996), H_∞ control (Williams et al,1989), LQG technique with loop transfer recovery (Baumgartner et al, 1986).

In most of these works the controller parameters are fine tuned using different optimization techniques and the final controller design is simplified and implemented in a classical PID format.

Except for the next three studies all the previous studies depend on a linear approach to the problem. Although with the increasing use of modern control techniques better performance is achieved but still the problem of disability to handle nonlinearities exists. As mentioned before the linearity assumption can be deteriorated if the plant dynamics exceeds the limits of linearization. Most of the linear controllers designed in the previous works perform badly outside the limits of its operating point.

Other unconventional approach such as fuzzy control and neural networks have been applied as in Abat and Dosio (1990), and Feldkamp and Puskorius (1992, 1993), where different computational techniques were applied aimed at increasing the controller robustness and decreasing the number of rules.

The aim of decreasing the number of rules reduces the memory requirements of the hardware but may cause discontinuities of the control surface and waste the advantage of existence and uniqueness of the solution in a closed loop system with fuzzy controller (Lewis, 1995). In this work it is noticed that the robustness and the range of operation of the control system is increased with increased number of rules.

Nonlinear black box modeling if done well and carefully planned can compensate well for different modeling uncertainties. In this work a nonlinear black box model will be developed and validated with correlation tests and a nonlinear fuzzy controller will be developed based on a full computational approach to avoid trial and error techniques used in the design of fuzzy controllers.

1.4 Objective of Research

The main goal of this research work is to develop a nonlinear black box mathematical model for a four cylinder port fuel injected engine idle mode that can be used for simulation and control purposes. The second goal is to develop a fuzzy controller based on a completely mathematical approach to stabilize the engine in its idle mode and compensate for different loads disturbing the engine. The controller must be robust enough to withstand abrupt changes of the engine parameters due to aging and unit to unit variability.

1.5 Contributions

Major contribution of this study can be listed as below

- Practical design of the identification experiment to collect data from the engine.
- Development of a multi input multi output nonlinear black box model for the idle mode using radial basis function network.
- Validating the model using nonlinear correlation tests.

• Development of a fuzzy controller for the idle mode based on a complete justified mathematical approach.

1.6 Organization of Thesis

The text of the thesis is organized in seven chapters including this chapter. Chapter 2 provides a background to nonlinear black box modeling and its relation to neural network. Chapter 3 discusses dynamic engine model and practical experimental setting for collecting the data used in the modeling. It presents the radial basis function network and its application to develop a nonlinear black box model for the idle mode where the problem is handled as a two input two output process and the network is trained by the orthogonal least squares algorithm. Chapter 4 is devoted to validate the model developed in chapter 3 by using nonlinear correlation tests. In chapter 5 a fuzzy controller is designed based on an optimum control table developed by the cell to cell mapping concept and trained by the orthogonal least squares algorithm. Simulation of the designed controller is demonstrated in the same chapter. Chapter 6 summarizes the research and the conclusion, in addition to suggesting future research regarding the subject.

CHAPTER 2

BACKGROUND TO ARTIFICIAL NEURAL NETWORK AND NONLINEAR SYSTEM IDENTIFICATION

2.1 Introduction

The purpose of this chapter is to describe briefly artificial neural networks (ANN) and its application to the system identification of nonlinear processes. First, neural network is quickly explored and the system identification problem is defined. Extension of linear model structure conventions to the nonlinear case is then treated and the problem is treated as a nonlinear function approximation problem in an ANN framework. This is followed by a discussion of networks architecture and modeling requirements.

2.2 Artificial Neural Networks

ANN has been used successfully in a wide range of engineering applications such as signal processing, image processing, pattern recognition....etc. One of its important and popular application is system control and modeling, and this is mainly due to its ability to handle nonlinear problems. For a survey of ANN in control see Hunt et al (1992) and for modeling see Sjoberg et al (1995). In the following we will treat ANN as nonlinear function approximators which is the main reason for using them in modeling.

Definition of ANN:

ANN models are algorithms for cognitive tasks, such as learning and optimization, which are in a loose sense based on concepts derived from research into the nature of the brain, (Muller and Reinhardt, 1991).

In general ANN are called connnectionist systems as it is a graph composed of interconnected nodes (neurons in network terminology). Mathematically, an ANN is a directed graph with the following properties (Figures 2.1, 2.2 and 2.3):

(1) An input vector x_i at time instant i.

(2) A defined transfer function $\kappa(.)$ associated with each node:

$$\kappa(.) = (x_i, \beta_k, \gamma_k) \qquad (2.1)$$

where each $\kappa(.)$ is a nonlinear function called the basis or the activation function and is considered as a node in the graph.

(3) A scale or directional property parameters, β_k of the transfer function κ.

(4) A position or translation parameters γ_k of the function κ.

In general there are three methods to construct $\kappa(.)$:

(a) The tensor product where the output of the node is the multiplication of activation function output of every dilated input alone.

$$\kappa_1(x_1) \cdot \kappa_2(x_2) \ldots \kappa_n(x_n) \qquad (2.2)$$

(b)The radial construction where the input to the activation function is the translation parameter (for example centers of the radial basis function) subtracted from the dilated input.

$$\kappa(x_i, \beta_k, \gamma_k) = \kappa(\| x_i - \gamma_k \|_{\beta_k}) \qquad (2.3)$$

(c) The ridge construction, Figure 2.2 where the input to the activation function is the sum of the dilated inputs and the translation parameter

$$\kappa(x_i, \beta_k, \gamma_k) = \kappa(\beta x_i + \gamma_k) \qquad (2.4)$$

The whole graph or mapping can be expressed as

$$\phi(x_i) = \sum \alpha \kappa(x_i, \beta_k, \gamma_k) \qquad (2.5)$$

where α is the coordinate parameters and κ can be a series of nested functions depending upon the architecture of the network.

Usually an ANN is referred to by the type of the basis function structure. For example sigmoidal network (Figure 2.2) has a feedforward structure with a sigmoidal activation function and the ridge structure. Radial basis networks has a feedforward architecture and a radial structure in the hidden layer. Other types are for example B-splines network, wavelet networks,...etc. On the basis of their architecture (Figure 2.3) an ANN can be a feedforward network (static), recurrent or feedback network (dynamic) or self-organizing network.

Networks used in system identification are mainly nonlinear function approximators. The basis for treating them as nonlinear function approximators is the Stone-Weierstrass theorem or the Kolmogorov's theorem.

The Kolomogorov 's theorem states that:

If I is the interval [0,1] and $I^n = [0,1]^n$ is a closed unit interval ($n \geq 2$), the Cartesian product of I. The theorem states that:

Any continuous mapping $NN = f(x_1,x_2,.....,x_n)$ of several variables defined on the space $I^n (n \geq 2)$ can be expressed in the form

$$f(x) = \sum_{j=1}^{2n+1} \Phi_j \left(\sum_{j=1}^{n} \Psi_{i,j}(x_i) \right) \qquad (2.6)$$

where Φ_j and is $\Psi_{i,j}$ are continuous functions of single variable and $\Psi_{i,j}$ are monotonic functions which are independent of f.

For network applications and as stated in Funahashi (1989), any continuous mapping $f: R^n \longrightarrow R^m$ defined by:

$$f(x)=(x_1,x_2,.....,x_n) \longrightarrow (f(x_1),f(x_2),.....,f(x_m))$$

can be approximated in the sense of uniform topology on R^n by input-output mapping of k-layer (k-2 hidden layers , $k \geq 3$, where k is the number of layers) network whose output functions for hidden layers are $\Phi_j(x)$, and whose output functions for input and output layers are linear.

In conjunction with the previous discussion and the system identification problem definition in section (2.3) the function approximation problem can be defined as follows:

Given $NN = f(x)$, a continuous function defined on a set X and an approximating function $\phi = F(\theta, X)$, that depends on a set P of parameters and X ,determine the vector of parameters, θ ($\theta \in P$,where θ is the parameter vector which combines α, β, γ) such that

$$d[F(\hat{\theta}, X), f(x)] \leq d[F(\theta, X), f(X)] \qquad (2.7)$$

for all θ in the set P, where $\hat{\theta}$ is the estimated vector of parameters and d is the distance function. The problems associated with this approximation definition are related to the

choice of suitable $F(.)$ for approximating the function f, which approximation schemes to use (sigmoidal, radial,...etc) and which algorithms (back propagation, recursive prediction error ...) for finding the optimal vector of parameters $\hat{\theta} \in P$ that minimizes the criterion associated with the function approximation scheme.

Once the approximation scheme has been decided on, a parameter adaptation algorithm based on one of the search for the optimum-algorithms has to be chosen ie Equation(2.8).

The process of training (or parameter estimation) is based on minimizing the sum of the square errors (over the number of input - output data points) between the estimated or approximated mapping and the real mapping, which is a nonlinear optimization process based on the following adaptation scheme

$$\hat{\theta}(i+1) = \hat{\theta}(i) - \alpha \, R_i^{-1} \, \nabla \hat{f}_i \qquad (2.8)$$

where:

α = step length.

R_i^{-1} = search direction (Newton, Gause-Newton, Levenberg-Marquardet,...).

$\nabla \hat{f}_i$ = the gradient.

How does an ANN work?

In most ANN applications the computation is done in two phases: the phase of training and the phase of recall or generalization. During training, the network tries to learn the object function through both the input and the corresponding output, whilst in generalization the network tries to recall the output corresponding to a given input. The ability of the network to learn depends on the input (properties, type and number of inputs), the type of activation function (some learn better than others), and the number

of activation function(some of these points will be discussed in later chapters in relation to system identification).

Training the network is an iterative process. Every network node combines the dilated translated inputs then, input the sum to the activation function to infer an output usually in the interval [0,1]. The output of the nodes is again dilated and translated or dilated and compared with the object function to calculate the error. Based on the condition and value of that error an adaptation or learning rule is used for adapting the network parameters to produce the optimum output. The training process is stopped when the error goal or a specific criterion is achieved (usually the average sum of square errors between the real and approximated function). After training we have matrices and vectors for both the dilation and the translation parameters which are used later for the generalization purpose.

In generalization, the network is given an input never given to it before and it is required to produce the optimum output corresponding to this input.

There is a trade - off between training accuracy and generalization accuracy. Strict accuracy condition requires excessive training leading to overfitting or overtraining which in turn causes a biased output in generalization. Training accuracy is dependent upon increasing the number of basis function i.e. number of nodes while generalization accuracy increases with decreasing the number of nodes (this will be discussed under the architecture issue).

2.3 Nonlinear System Identification and Networks

Here the problem of system identification and its relation to ANN is defined for modeling purpose.

2.3.1 Problem definition

The system identification problem is defined as follows:

Given a series of input-output observations from a dynamical system

$$u^t=[u(1)\ \ u(2).....u(t)] \qquad (2.9)$$

$$y^t=[y(1)\ \ y(2)\ .....y(t)] \qquad (2.10)$$

find a parameterized model structure to predict the future behavior of the system (predicted outputs $y(t)$). Generally this is done by introducing a family of nonlinear finitely parameterized functions and searching one of these functions as the candidate model structure. Defining the parameters by θ and the nonlinear function by $g(.)$ then

$$y(t) = g(u^{t-1},y^{t-1},\ \theta) + v(t) \qquad (2.11)$$

where $v(t)$ represents the residuals of the model (the dynamics which cannot be caught by the model).

Usually u^t,y^t are combined in one vector φ called the regressors vector. The general equation takes the form:

$$y(t) = g(\varphi,\ \theta) + v(t) \qquad (2.12)$$

A good model is one which has minimum $v(t)$, i.e. the model in which the real output $y(t)$ can be represented approximately by the function $g(.)$ ie $y(t) \sim g(\varphi,\underline{\theta})$. The main problems in fitting Equation (2.12) to the real data are choosing the appropriate nonlinear function $g(.)$ (model structure), the regressor vector φ for the best fit between the estimated model and the real process. The next step is to find the vector of parameters estimates $\hat{\theta}$, which gives the best fit between the model Equation (2.12) and

the output $y(t)$, and this is done through an optimization process aiming at minimizing the following criterion: $J=\sum_{1}^{N} \| y(t) - g(u^{t-1}, y^{t-1}, \theta) \|^2$

2.3.2 Extension of the linear case to the nonlinear case

The general form of the linear system model structure(input-output model) is given by

$$A(q)y(t) = B(q)/F(q)\ u(t-n_u) + C(q)/D(q)\ e(t) \qquad (2.13)$$

where q is the shift operator (z^{-1} may be used also), A , B , C and F are polynomials of q^{-1} , $e(t)$ is the noise or residuals term(can be white noise) and n_u is the time delay. Many variations can be done to this general form to give various model structures such as ARX model ($F(q)= C(q) = D(q) = 1$), ARMAX ($F(q)=D(q)=1$),impulse response model ($A(q)= F(q)= C(q) = D(q) = 1$),output error model ($A(q)=C(q)=D(q)=1$) and the Box Jenkins (BJ) model . The order of the model (number of poles) is defined by the order of polynomial A, the number of zeros is defined by the order of B.

The regressors used in the general form of Equation (2.13) are:

$u(t-k)$ with the polynomial B

$y(t-k)$ with the polynomial A

$\hat{y}(t-k)$ with the polynomial F

$e(t-k) = (y(t-k) - \hat{y}(t-k))$ with the polynomial C

$e_u(t-k) = y(t-k) - \hat{y}_u(t-k)$ with the D polynomial.

The general form of the linear state space model is given by :

$$x(t+1) = A\,x(t) + B\,u(t) + K\,e(t)$$

$$y(t) = C\,x(t) + D\,u(t) + e(t) \qquad (2.14)$$

where A , B , C , D are matrices and $e(t)$ is the noise term.

By extending these concepts to the nonlinear case we can get the following variants (Sjoberg et al.(1995) and Billings et al (1988)):

[1] Input output forms

(a) NFIR (Nonlinear finite impulse response) models with *u(t-k)* as the regressors.

(b) NARX models with *u(t-k)* and *y(t-k)* as the regressors(series-parallel model) .

(c) NARMAX models with *u(t-k)*and *y(t-k)* and *e(t-k)*as the regressors .

(d) NOE models with *u(t-k)* and $\hat{y}$ *(t-k)* as the regressos(parallel model) .

(e) NBJ models with *u(t-k)* , $\hat{y}$ *(t-k)* , *e(t-k)* and $e_u(t-k)$ as the regressors.

[2] Nonlinear state space models (Nerrand et al(1994), Nerrand et al(1993)):

(a) State space OE model

$$x(t+1)=f[x(t)),u(t)]$$

$$y(t)=x(t)+v(k)$$

(b) State space NARMAX model

$$x(t+1)=f[x(t),u(t),v(t)]+v(t)$$

$$y(t)=x(t)$$

(c) State space NARX model

$$x(t+1)=f[x(t)),u(t)]+v(t)$$

$$y(t)=x(t)$$

v(t) is the noise term , *x(t-1)* is the state vector and *u(t)* is the input.

2.4 Formulation of System Identification Problem

In this section the formulation of the system identification problem in discussed in a network framework. The ANN is viewed as a nonlinear mapping ($g(\varphi,\theta) = \sum\alpha_k g_k(\varphi)$) approximators and its relation to system identification is presented.

As mentioned before, difficulties associated with the system identification problem are mainly concerning the nonlinear function $g(.)$, the parameters vector θ and the regression vector φ ,which define the model type (NARX, NARAMAX,.....).The nonlinear mapping $g(.)$ can be viewed as the function expansions :

$$g(\varphi, \theta) = \sum \alpha_k \, g_k(\varphi) \qquad (2.15)$$

where g_k 's are called basis function (corresponds to model structure and may be sigmoidal, radial basis, B splines,.....), g_k 's are obtained by parameterizing a mother basis function (Sjoberg et al,1995), $\kappa(.)$ as follows:

$$g_k(\varphi) = \kappa(\varphi, \beta_k, \gamma_k) \qquad (2.16)$$

β_k is the dilation or the scale parameter (weight matrix in ridge construction type($g_k(\varphi) = \kappa(\beta_k^T \varphi + \gamma_k)$) , γ_k is a translational parameter (bias in ridge construction, centers in radial construction($g_k(\varphi) = \kappa(\| \varphi - \gamma_k \| \beta_k)$) and α_k are coordinate parameters.

The basis function expansions, Equation (2.15) which correspond to the model structure can be illustrated graphically in the form of a feedforward network as can be seen in Figure 2.1. Here the mother basis function $g_k(\varphi) = \kappa(\varphi, \beta_k, \gamma_k)$ is illustrated as a node and repeated for a number of times. The number of nodes has to be determined and corresponds to the number of neurons in the network. The number of parameters in the identified model equation is dependent on the number of neurons, θ now includes the whole set of the network parameters and the model structure (NARX, NARMAX,...) is determined by the regressors which are the inputs to the network.

Except for NFIR and NARX, other model structures correspond to the structure of recurrent networks as parts of the regression vector are past outputs of the model.

Training the network corresponds to the process of parameter estimation (finding the optimum values for the parameters θ, based on Equation (2.8)), while generalization corresponds to model validation.

2.5 Network Architecture and Modeling Requirements

System identification mainly concerns two things: the first is the model structure identification and the second is the parameter set identification. In this section some general topics related to neural network architecture are discussed in relation to parameter set identification. This is closely related to model quality measures, namely bias and variance of estimation and their relation to the architecture of the network.

By quality of the model we mean the ability of the model to catch (from the estimation data set) and reproduce (from the validation data set) the dynamics of the real process. Catching the whole dynamics of the model is a matter of having an efficient fitting procedure, whilst reproduction of the dynamics is a matter of having a suitable model structure in conjunction with efficient fitting procedure. Model quality which corresponds to the fitting procedure is closely related to network architecture where the approximation potential of the network depends strongly on the number of the basis functions in the hidden layer (nodes). On the other hand model quality aspect which corresponds to model ability to reproduce the dynamics of the system is dependent upon the input to the network, in other words the dimension of the regressor vector.

It is generally known that model quality can be studied by analyzing the residuals of the model estimation process. By residuals of a model we mean the

prediction error $\varepsilon(t) = y(t) - \hat{y}(t,\hat{\theta},\varphi)$ where $y(t)$ is the real process output and $\hat{y}(t,\hat{\theta},\varphi)$ is the estimated model output of the process.

The residuals evolve from two different sources; the first one is the noise that contaminates the measured data of the process, whilst the second source comes from the process of constructing the mathematical model from even noise - free data. Residuals evolving from the first source are known as variance errors whilst residuals from the second source are known as bias errors. Residuals of the first type can be theoretically made negligible by using an infinite series of data points $N \to \infty$ whilst the second one can be overcome by using suitable model structure and parameter set.

In terms of the its parameters a good model is judged by the criterion

$$min \quad \bar{V}(\theta) = E\{(y(t) - g(\varphi(t),\theta)(y(t) - g(\varphi(t),\theta)^T\} \qquad (2.17)$$

$\bar{V}(\theta)$ represents the covariance matrix of the parameters, $g(\varphi(t),\theta)$ represents the estimated model , $\varphi(t)$ are the regressors and θ is the set of parameters.

Assuming that the measurement noise has a variance λ then

$$\bar{V}(\theta) = \lambda + E\{(g_o(t) - g(\varphi(t),\theta)(g_o(t) - g(\varphi(t),\theta)^T\} \qquad (2.18)$$

In practice the criterion $\bar{V}(\theta)$ can be represented as the sample mean

$$\bar{V}(\theta) = \frac{1}{N}\sum_{t=1}^{N} [(y(t) - g(\varphi(t),\theta)(y(t) - g(\varphi(t),\theta)^T] \qquad (2.19)$$

Let $\theta_*(m)$ represents the optimum parameter set, then the best model can be represented by:

$$\theta_*(m) = arg\{ min\ \bar{V}(\theta) \} \qquad (2.20)$$

where m is the dimension of the parameters vector (which is proportional to the number of basis functions). A measure of the model quality which describes the model fit when applied to a new data set is:

$$E\{\bar{V}(\hat{\theta})\} = V_*(m) \qquad (2.21)$$

Based on the assumption of white measurement noise Equation (2.13) can be decomposed into three basic terms

$$\bar{V}_*(m) = \lambda + E(g_o(\varphi) - g(\varphi(t),\theta_*(m))(g_o(\varphi) - g(\varphi(t),\theta_*(m))^T$$

$$+ E(g(\varphi,\theta_*(m)) - g(\varphi,\hat{\theta}_N))(g(\varphi,\theta_*(m)) - g(\varphi,\hat{\theta}_N))^T \qquad (2.22)$$

where $\hat{\theta}_N$ is the parameter estimate from the estimation data set.

The second and third terms of Equation (2.22) respectively represent the bias and variance of estimation. As the number of data points N tends to infinity the variance of estimation will be negligible and the model parameters θ_N, will tend to the best or optimum parameter set θ_*.

$$\hat{\theta}_N \to \theta_*(m) = arg \min_{\theta} \bar{V}(\theta) \qquad (2.23)$$

which implies

$$\frac{1}{N}\sum_{t=1}^{N} (y(t) - g(\varphi(t),\theta)(y(t) - g(\varphi(t),\theta)^T \to E\{\varepsilon^2(t)\} = \bar{V}(\theta) \qquad (2.24)$$

Under the assumption of predicted white noise and zero bias term the variance term in Equation (2.22) can be treated as follows (Ljung,1994)

$$E(g(\varphi,\theta_*(m)) - g(\varphi,\hat{\theta}_N))(g(\varphi,\theta_*(m)) - g(\varphi,\hat{\theta}_N))^T \approx \frac{1}{N}\lambda R^{-1} \qquad (2.25)$$

R^{-1} represents the parameter covariance matrix $R = E\psi(t,\theta_*)\psi(t,\theta_*)^T$ and

$$\psi(t,\theta_*) = \frac{d}{d\theta}\hat{y}(t,\theta) \qquad (an\ m \times 1\ vector)$$

From Equation (2.25) it can be concluded that the uncertainty in a component of θ is related to the sensitivity of the model $\hat{y}(t,\theta)$ to a change in that component. In other words, a small change of $\hat{y}(t,\theta)$ for a change in component of the parameters θ will produce a small change in $\overline{R}$, consequently a large uncertainty of θ. In Sjoberg et al (1995) Equation (2.25) is approximated to

$$E\{(g(\varphi,\theta_*(m)) - g(\varphi,\hat{\theta}_N))(g(\varphi,\theta_*(m)) - g(\varphi,\hat{\theta}_N))^T\} \approx \frac{1}{N}\lambda m \qquad (2.26)$$

then Equation (2.22) can be expressed as

$$\overline{V}_*(m) = \lambda + \frac{1}{N}\lambda m + E(g_o(\varphi) - g(\varphi(t),\theta_*(m))(g_o(\varphi) - g(\varphi(t),\theta_*(m))^T \qquad (2.27)$$

$$\overline{V}_*(m) = \lambda + \frac{1}{N}\lambda m + \overline{V}(\theta_*(m)) \qquad (2.28)$$

Equation (2.27) describes the expected loss when the model is applied to a new data set. It also describes the effect of changing the number of parameters m, (changing network architecture directly changes m) on the loss function of Equation (2.26). In Equation (2.26), $\overline{V}_*(m)$ is a nonincreasing function of m. Increasing the number of basis functions (which implies increased m) increases the approximation potential of the network, and hence directly increases the estimation variance term and unless the

decrease in $\overline{V}_*(m)$ is not less than λ / N the increase of m will increase the importance of the unimportant parameters and severely affects the overall quality criterion $\overline{V}_*(m)$ of the model leading to overfitting. The number of parameters is proportional to the number of basis functions used in the hidden layer. It therefore depends on network architecture. Increasing the number of basis function in seeking for more approximation capacity results in increasing the number of parameters, consequently leading to overfitting.

In the case of the adaptive model both the second and third terms of Equation (2.22) are considered, while all the terms are considered in relation to the radial basis function model case. In the radial basis model case the error tolerance ratio ζ controls the structure of the network and consequently the complexity of the model. Very small values of ζ increase the number of hidden layer nodes and small values decreases the number of these nodes.

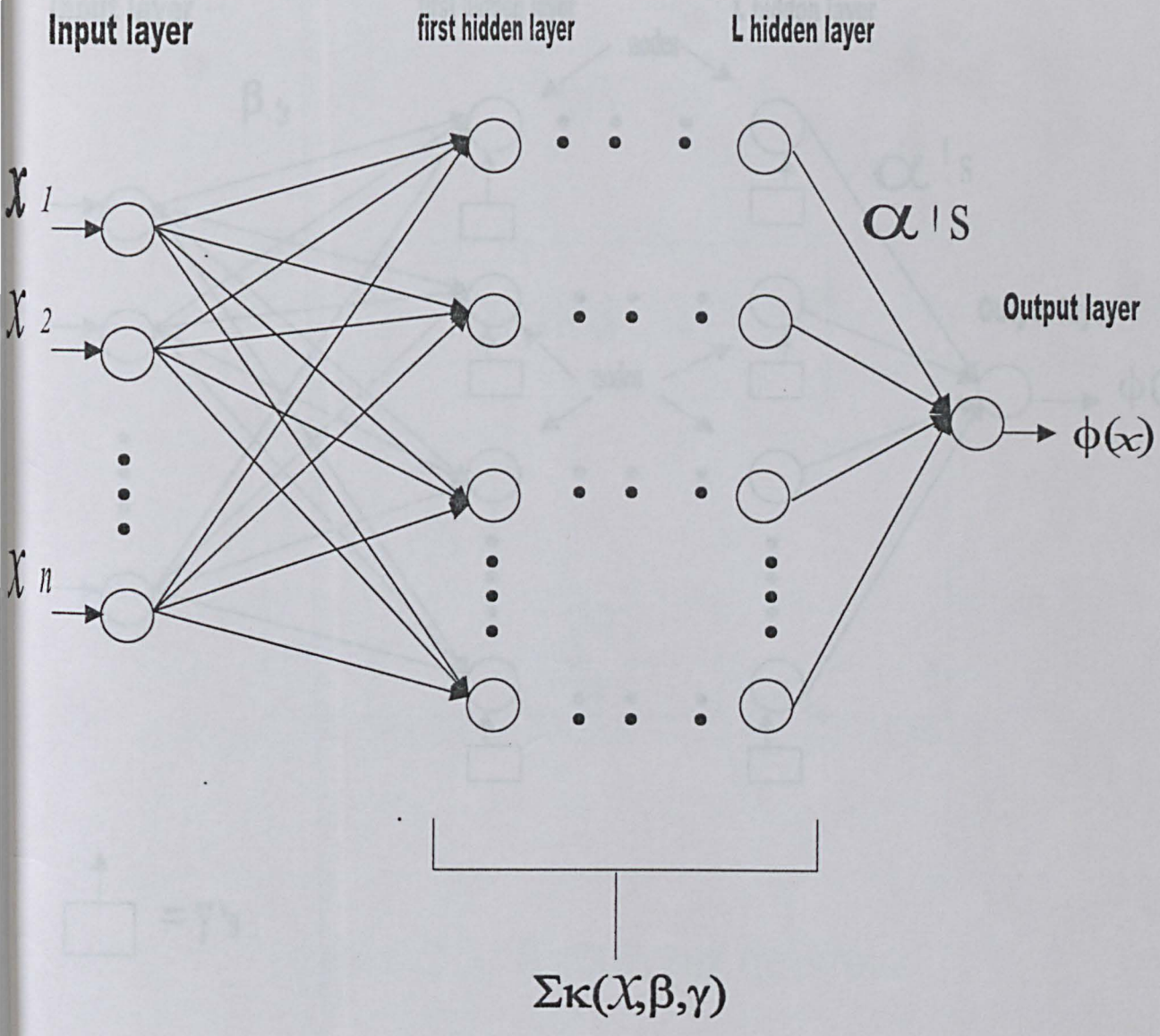


Figure 2.1 Artificial Neural Network

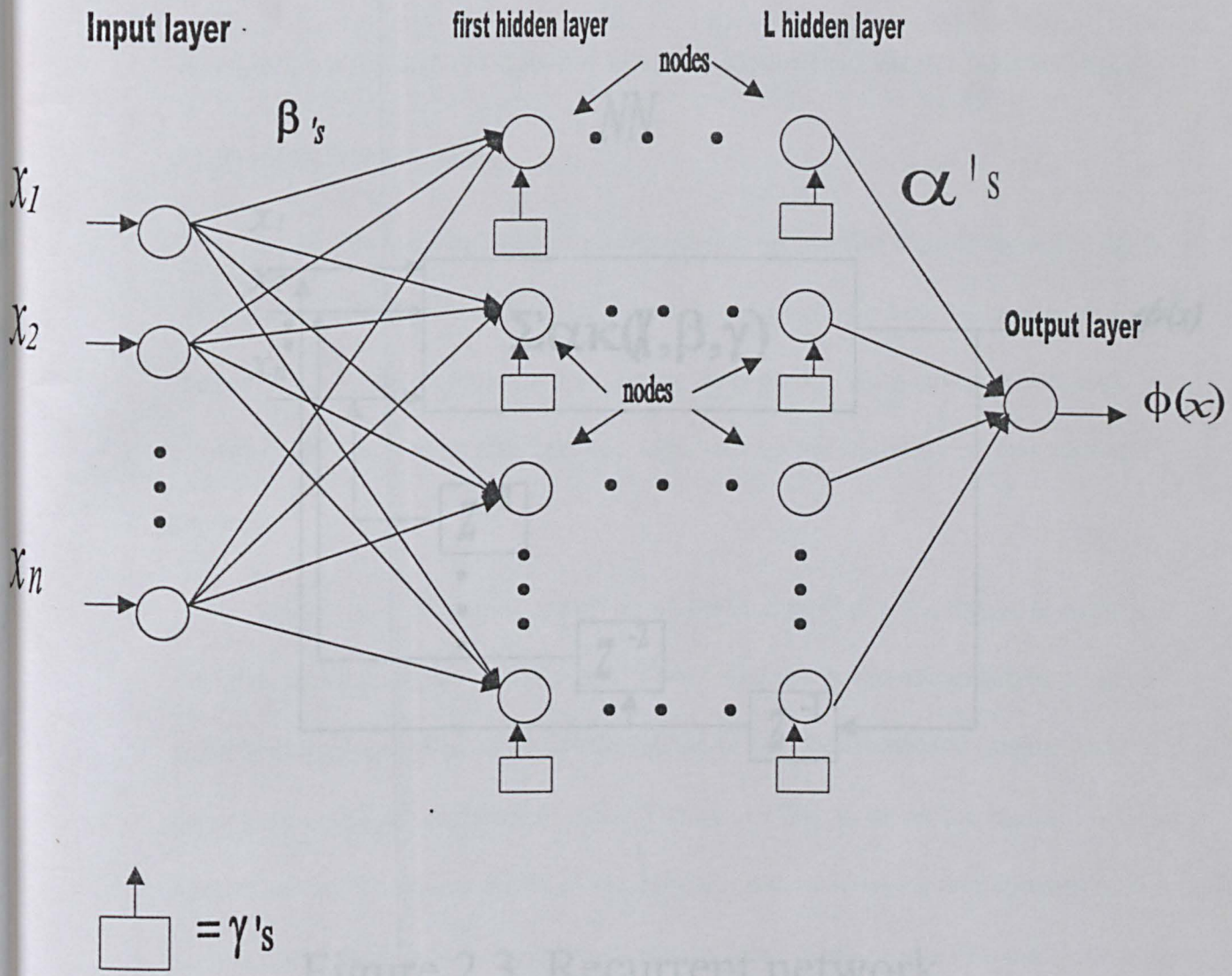


Figure 2.2 Sigmoidal Network

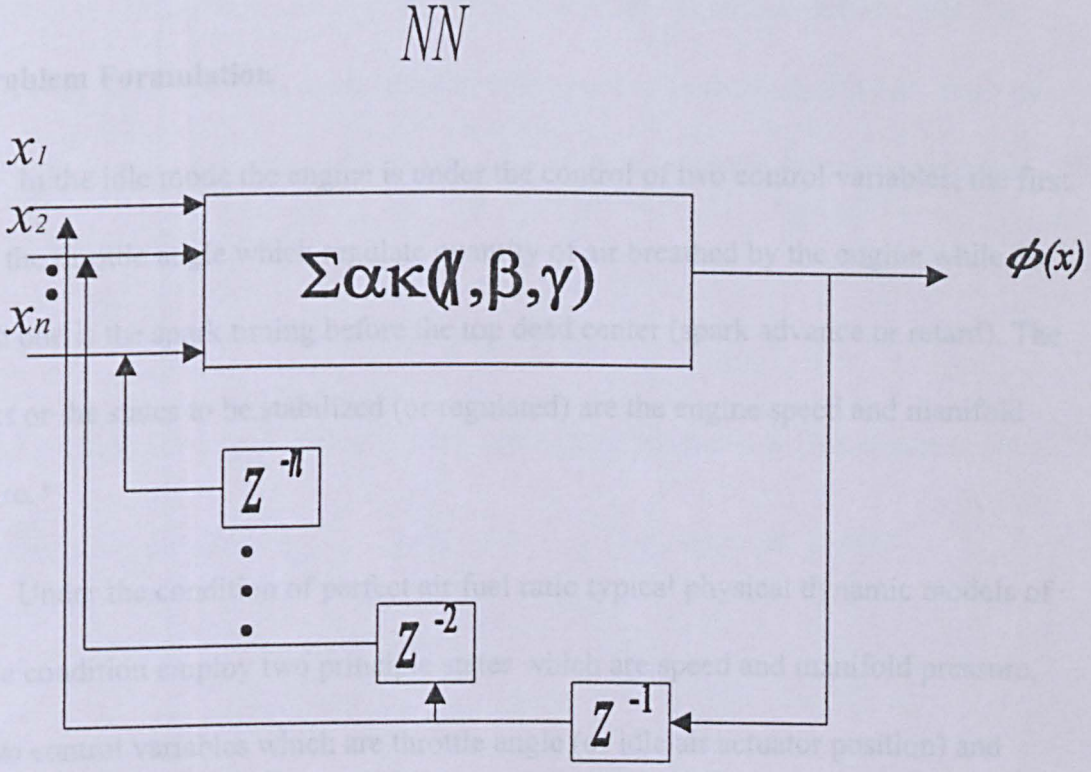


Figure 2.3 Recurrent network

CHAPTER 3

ENGINE MODELING USING RADIAL BASIS FUNCTIONS NETWORKS

3.1 Problem Formulation

In the idle mode the engine is under the control of two control variables, the first one is the throttle angle which regulate quantity of air breathed by the engine while the second one is the spark timing before the top dead center (spark advance or retard). The outputs or the states to be stabilized (or regulated) are the engine speed and manifold pressure.

Under the condition of perfect air fuel ratio typical physical dynamic models of the idle condition employ two principle states which are speed and manifold pressure, and two control variables which are throttle angle (or idle air actuator position) and spark timing [Morris and Powel (1983), Powell (1995)], to model the engine. Different approaches to idle mode modeling have been in existence (see the references of section 1.3). Some of them handle the process as single input single output (throttle is the input and speed is the output). Others handle it as two inputs one output where throttle angle and ignition timing are the inputs and speed is the output. In this work the process is handled as a multi-input multi-output (MIMO) process where including the manifold pressure in the model helps in making the idle speed more stable. The manifold pressure is sensitive to the ignition advance and throttle angle, pressure overshoot severely affects the speed dynamics. This can be shown from simulating the model under different initial conditions of the pressure.

In black box modeling a process is viewed as a black box and attention is only paid to both the output and input variations irrespective of what is going on inside the process. Keeping this principle in mind we construct a nonlinear black box dynamical model for the idle mode of our engine. The inputs observed are the throttle angle and spark timing while the outputs observed are the speed and manifold pressure. Both the outputs of the process are interacting with each other to yield the following nonlinear discrete mathematical form:

$$y(k+1) = f(y(k), u(k))$$

where $y(k+1)$ is the output vector $f(.)$ is a vector function and $u(k)$ is the control vector.

$y(k+1)=[\, y_1(k+1)\ \ y_2(k+1)\,]^T$, $f=[f_1\ \ f_2]^T$, $y(k)=[\, y^t{}_1(k)\ \ y^t{}_2(k)\,]^T$

$u(k)=[\, u^t{}_1(k)\ u^t{}_2(k)]^T$ The control vector

$y_1(k+1)$ engine speed

$y_2(k+1)$ manifold pressure

$y^t{}_1(k)=[\, y_1(k-1), \ldots, y_1(k-n_y)]$

$y^t{}_2(k)=[\, y_2(k-1), \ldots, y_2(k-n_y)]$

$u^t{}_1(k)=[\, u_1(k-1), \ldots, u_1(k-n_u)]$ throttle angle

$u^t{}_2(k)=[\, u_2(k-1), \ldots, u_2(k-n_u)]$ spark timing

where n_y and n_u represent both the number of delayed outputs and inputs involved in fitting the dynamics of the model.

With the principle of black box model we handle the engine as a black box injected with two inputs u_1 ,u_2 and producing two outputs y_1 ,y_2 . A series of random input output signals are injected to the plant and the two outputs are observed and recorded.

3.2 Radial Basis Functions Networks

In using networks for system identification two approaches can be used for determining the parameters of the network α,β and γ. The first approach is to estimate α,β and γ simultaneously using suitable training algorithm (e.g. back propagation, error prediction algorithm,...). This approach yields a highly nonlinear in the parameters network model which results in a nonlinear learning rule leading to the problem of local minima.

In the second approach which is called constructive approach (Sjoberg et al, 1995), the network parameters α and β are fixed (for example with suitable clustering technique ART2 or K means clustering). Thus estimation of the coordinate parameter α 's can be done with a linear regression algorithm. In this way we avoid the local minima and overtraining troubles mentioned previously. Radial basis function network with radial structure in the hidden layer is one of the networks that can be used with the second approach to approximate nonlinear functions.

Radial basis function networks (RBF) are two layer networks with nonlinearities of fixed parameters in the hidden layer and adjustable parameters in the output layer which are adjusted by a linear regression algorithm. The approximation capabilities of the RBF networks stems from its localization properties as the gradient of the function is within bounded support or at least vanishing rapidly at infinity.

RBF networks have the structure

$$f(\varphi,\alpha,\beta) = \alpha_O + \sum_{i=}^{nh} \alpha_i g(\| \varphi - \gamma \|_\beta) \qquad (3.1)$$

where

α_o=constant.

g = nonlinearity.

φ = input x (the regressors).

α_i = coordinate parameters.

γ =dilation parameters or centers of the RBF which are some fixed points in the n-dimensional space sampling the input domain.

$\| \cdot \|_\beta$ = any norm (usually the quadratic norm).

n_h = number of hidden layers.

Nonlinearities in the hidden layers can be taken to be one of the following nonlinearities

-The thin plate function. $g(x) = x^2 \log(x)$ (3.2)

-The Gaussian function. $g(x) = e^{x^2/\sigma}$ (3.3)

-The multiquadratic function. $g(x) = (x^2 + \sigma^2)0.5$ (3.4)

-The inverse multiquadratic function. $g(x) = (x^2 + \sigma^2)- 0.5$ (3.5)

The chosen nonlinearity does not affect the approximation capabilities of the function but may affect the numerical properties of the algorithm and this is especially so with the gaussian function.

The main problem of the RBF network is the determination of the centers while, coordinate parameters can be determined by a linear regression algorithm. When RBF networks were first introduced these centers were taken as the whole data set or some centers were taken randomly from the data set. This proves to be unpractical especially for large input space. Chen et al (1989) derived a feedforward algorithm called the orthogonal least squares algorithm (OLS) for subset model selection. This algorithm is then used to train RBF networks to choose the centers of the RBF network from the collected input output data points in conjunction with the determination of the coordinate parameters α_i. The authors viewed the problem of choosing centers as choosing a set of subset regressors from an extended set of regressors.

In the following sections the OLS algorithm will be used to train an RBF to find a two state two input equation error model (NARMAX) for modeling the engine idle mode.

3.3 The Orthogonal Least Squares Algorithm

OLS algorithm mainly selects the basis function (which are formed by the whole set of regressors), that best fit the estimated data then, repeatedly selects the basis function from the remainder of the set of basis functions while combining and orthogonalizing with the earlier selected basis functions. In this way, a group of radial basis functions which can span the input space is obtained.

Assuming that the centers of the RBF 's are chosen as the whole data set with each center having the dimension of the input regressor vector x, then RBF network can be viewed as a linear regression model of the form

$$y_i(t) = \sum_{j=1}^{M} g_j(t)\,\theta_{ji} + e_i(t) \qquad 1 \le i \le m \tag{3.6}$$

where $y_i(t)$ is the ith output signal and m is the number of outputs.

$$g_j(t) = [\,g_j(1), g_j(2), \ldots\ldots g_j(N)\,] \qquad 1 \le j \le M \tag{3.7}$$

$g_i(t)$ is the *ith.* nonlinear basis function.

$$y_i(t) = [\,y_i(1), y_i(2), \ldots\ldots y_i(N)\,] \qquad 1 \le i \le m \tag{3.8}$$

$e_i(t)$ is the ith residual of the model

$$e_i(t) = [\,e_i(1), e_i(2), \ldots\ldots e_i(N)\,] \qquad 1 \le i \le m \tag{3.9}$$

$e_i(t)$ is assumed to be uncorrelated with the regressors.

θ_{ji} are the parameters.

In Equation (3.6) we notice that $g_j(t)$'s are equivalent to the regressors of a linear regression model. $g_j(t)$'s are some fixed basis functions of the input x, i.e. $g_j(t) = g_j(x(t))$. It has the dimension N which is equal to the number of data points.

For the MIMO case Equation (3.6) can be rewritten as

$$[\,y_1 \ldots, y_m\,] = [g_1, \ldots g_m] \begin{bmatrix} \theta_{11} & \ldots & \theta_{1m} \\ \theta_{21} & \ldots & \theta_{2m} \\ \vdots & \ldots & \vdots \\ \theta_{M1} & \ldots & \theta_{Mm} \end{bmatrix} + [e_1, e_2, \ldots e_m] \tag{3.10}$$

which can be written as

$$Y = G\Theta + E \qquad (3.11)$$

Equation (3.11) has an interesting geometric interpretation; the regressor vectors g_i form a set of basis vectors which span the output space Y, and consequently the set g_i must be an orthogonal set to specify the contribution of each g_i to the output Y. The parameters $\hat{\Theta}$ satisfy the condition that Y is the projection onto the space spanned by the set g_i.

The columns of the regession matrix are always nonorthogonal and the OLS aims at orthogonalizing the matrix to yield a compact set of basis vectors which can span the output space Y. In other words, this is contribution of each individual regressor to the output energy. From the above representation it can be concluded that the center selecting process is equivalent to selecting a subset of regressors from a given set of regressors. The regressor matrix G can be decomposed into:

$$G = W B \qquad (3.12)$$

where

$$W = [\, w_1 \;, w_2 \;, \ldots\ldots, w_M \,] \qquad (3.13)$$

is an orthogonal matrix of dimension $N \times M$ with the property that $w_i^T w_j = 0 \quad i \neq j$.

B is an $M \times M$ triangular matrix of elements equal to one on the diagonal .

$$B=\begin{bmatrix} 1 & \beta_{12} & \beta_{13} & . & . & . & \beta_{1M} \\ 0 & 1 & \beta 23 & . & . & . & \beta_{2M} \\ 0 & 0 & . & . & . & . & . \\ 0 & . & 0 & . & . & . & . \\ 0 & . & . & 0 & . & . & . \\ 0 & . & . & . & 0 & . & \beta_{M-1,M} \\ 0 & 0 & . & . & . & 0 & 1 \end{bmatrix} \qquad (3.14)$$

The set w_i span the same space spanned by the set G_i and consequently the system of Equation (3.11) is transformed to the following system:

$$Y = W B \Theta + E \qquad (3.15)$$

$$Y = W \Gamma + E \qquad (3.16)$$

where $$\Gamma = B \Theta \qquad (3.17)$$

W is called the auxiliary regressor matrix and Γ is called the auxiliary parameters matrix.

The OLS solution is then given by:

$$\hat{\Gamma}=\begin{bmatrix} \hat{\gamma}_{11} & \hat{\gamma}_{12} & . & . & \hat{\gamma}_{1m} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \hat{\gamma}_{M1} & . & . & . & \hat{\gamma}_{Mm} \end{bmatrix} \qquad (3.18)$$

$$\hat{\Gamma} = (W^T W)^{-1} W^T Y$$

$$\hat{\gamma}_{ji} = w_j^T y_i / w_j^T w_j \qquad 1 \le j \le M \qquad (3.19)$$

where the OLS solution $\hat{\Gamma}$ and the LS solution $\hat{\Theta}$ satisfy the triangular system

$$B\,\hat{\Theta} = \hat{\Gamma} \qquad (3.20)$$

The decomposition Equation (3.12) can be done with any of the decomposition methods (Gram-Schmidt, House Hold transformation,...). In this work here the Gram-Schmidt process (GS) will be used to determine this composition .The GS process is used to infer a set of orthogonal vector set (basis vectors) to span the vector space containing the set of this basis vectors. The problem is described as follows:

Given a set of vectors G, find a set of orthogonal vectors, W which can span the space G.The GS process is described as follows:

(1) Select a vector g_1 as the first basis vector

$$w_1 = g_1$$

(2) Select w_2 as the vector formed from g_2 by subtracting out the component of g_2 in the direction of w_1, this is equivalent to requiring that the inner product $<w_1\ w_2> = 0$

$$w_2 = g_2 - \beta_{12}\ w_1 \quad , \quad \beta_{12} \text{ is the first column of B}$$

The condition of orthogonality is

$$\beta_{12} = <w_1, g_2> / <w_1\ w_1>$$

$$w_3 = g_3 - \beta_{13}\ g_1 - \beta_{23} g_2$$

$$\beta_{13} = <w_1\ g_3> / <w_1\ w_1> \quad , \quad \beta_{23} = <w_2\ g_3> / <w_2\ w_2>$$

$$\beta_{ik} = \langle w_i^T g_k \rangle / \langle w_i^T w_i \rangle$$

$$w_k = g_k - \sum_{i=1}^{k-1} \beta_{ik} \; w_i \qquad k=2,.....M$$

ie one column of B is calculated at a time and the kth column is made orthogonal to each of the k-1 previously orthogonalized columns and the operation is repeated for M-1 times. The previous procedure leads to excessively increased number of parameters and consequently to a complex model structure.

As a few number of regressors $M_S << M$ can be enough to describe effectively the dynamic behavior of the system the OLS employs a criterion to limit the number of regressors to a compact number. This criterion is called the error reduction ratio, which is explained as follows:

The trace of the covariance of $y(t)$ is given by:

$$trace(Y^T Y/N) = \frac{1}{N} \sum_{j=1}^{M} \left(\sum_{1}^{m} \gamma_{ji}^2 \right) w_j^T \; w_j + trace\left(\frac{E^T E}{N} \right) \qquad (3.21)$$

The first right hand term of Equation (3.21) represents the explained part of the covariance of the desired output due to the regressors while the second part represents that part which is the unexplained part of the desired output. Thus, the first term is the increment to the explained trace introduced by w_i and the error reduction can be defined as given in Chen (1991):

$$[err]_i = \left(\sum_{1}^{m} \gamma_{ji}^2 \right) \; w_j^T \; w_j \; / \; trace\left(\frac{E^T E}{N} \right) \qquad 1 \le i \le M \qquad (3.22)$$

Equation (3.22) represents the contribution of the ith regressor w_i and consequently the ith regressor that maximizes the error reduction ratio. Centers which produce this ith regressor will thus be selected as candidate centers.

The algorithm steps can be represented as follows:

Step 1

For $1 \le i \le M$

$$w_1^i = g_i$$

$$\gamma_{1q} = (w_1^i)^T Y_q / (w_1^i)^T (w_1^i) \qquad 1 \le q \le m$$

$$[err]_1^i = \left(\sum_1^m (\gamma_{1q}^i)^2 \right) (w_1^i)^T w_j^i \Big/ trace\left(\frac{E^T E}{N} \right)$$

Find $[err]_i = [err]_1^{i1} = max \{ [err]_1^i , 1 \le i \le M \}$

$$w_1^{i1} = w = g_{i1}$$

Step 2

At the kth step where $k \ge 2$, for $1 \le i \le M \quad i \ne i_1 \quad ,\ldots\ldots\ldots\ldots\ldots\ldots, i \ne i_{k-1}$

$$\beta_{ik} = (w_j^T g_k) / (w_j^T w_j) \qquad 1 \le j \le k$$

$$w^i{}_k = g_i - \sum_{j=1}^{k-1} \beta_{jk}^i \; w_j$$

$$\gamma^i{}_{kq} = (w_k^i)^T Y_q / (w_k^i)^T (w_k^i) \qquad 1 \le q \le m$$

$$[err]^i{}_k = \left(\sum_1^m (\gamma^i{}_{kq})^2 \right) (w_k^i)^T w_k^i \Big/ trace\left(\frac{E^T E}{N} \right)$$

Find $[err]_k = [err]_{k^{ik}} = max \{[err]_k^i , 1 \le i \le M , i \neq i_1 , \ldots\ldots\ldots\ldots, i \neq i_{k-1} \}$

$$w_k = w^{ik}{}_k = g_{ik} - \sum_{j=1}^{k-1} \beta^i_{jk} \; w_j$$

$$1 - \sum_{j=1}^{M_s} [err]_j < \zeta \qquad 0 < \zeta < 1$$

The process is repeated until a predefined value for the summation of the error reduction ratios is reached. The value of the parameter ζ indirectly affects the complexity of the network (i.e. the model). It may increase the number of the hidden layer nodes leading to overparametrization of the model. In applying the RBF for the modeling process using the OLS algorithm, the RBF numerical properties are sensitive to two factors: the first one is the spread parameter of the Gaussian function, σ and the second one is the ill conditioned matrix W. The first factor σ can be selected by trial and error .The second factor is adjusted by introducing a threshold value to $w^T w$ where $w^T w = 0$, implying that the regressors $g(k)$ and $g(k-1)$ are linearly dependent.

3.4 Mathematical Form of the MIMO Nonlinear Model

The general expression for a MIMO nonlinear model structure is

$$y(t) = f(y^{t-1}, u^{t-1}, \varepsilon^{t-1}) + \varepsilon(t) \qquad (3.23)$$

where

$y(t) = [y_1(t), y_2(t), \ldots\ldots, y_q(t)]^T$ *The output vector*

$u(t) = [u_1(t), u_2(t), \ldots\ldots\ldots, u_r(t)]^T$ *The input vector*

$\varepsilon(t) = [\varepsilon_2(t), \varepsilon_2(t), \ldots\ldots\ldots \varepsilon_q(t)]^T$ *The residuals vector*

$$f = [f_1 , f_2 , \ldots\ldots f_r]^T$$

$$y^{t-1} = [\ y_1^{t-1} , y_2^{t-1} , \ldots\ldots , y_q^{t-1}]^T$$

$$u^{t-1} = [\ u_1^{t-1} , u_2^{t-1} , \ldots\ldots , u_r^{t-1}]$$

$$\varepsilon^{t-1} = [\ \varepsilon_1^{t-1} , \varepsilon_2^{t-1} , \ldots\ldots , \varepsilon_q^{t-1} \ ]$$

$y_i^{t-1} = [y_i\ (t-1)\ , y_i\ (t-2)\ , \ldots\ldots , y_i(t-n_y)\ ]$ *delayed output vector*

$u_i^{t-1} = [u_i(t-1)\ , u_i(t-2\ ) , \ldots\ldots , u_i\ (t-n_u)\ ]$ *delayed input vector*

$\varepsilon_i^{t-1} = [\ \varepsilon_i(t-1)\ , \varepsilon_i t-2) , \ldots\ldots , \varepsilon_i(t-n_\varepsilon)\ ]$ *delayed error*

In our case and for the sake of simplicity of the model a first order assumption is first demonstrated where, $q=2$ and $n_y=n_u= n_\varepsilon =1$ and the inputs are:

$u^{t-1} = [u_1^{t-1}\ \ u_2^{t-1}] = [\ T(k-1)\ \ D(k-1)\ ]$ where T is the throttle angle [degree]and D is the ignition advance[degree]. The outputs are:

$y(t) = [\ y_1(t)\ \ y_2(t)]^T = [\ N(k)\ \ P(k)]^T$ where $N(k)$ is the speed [r.p.m] and $P(k)$ is the manifold pressure [kpa].

The previous multi input model structure corresponds to the nonlinear NARMAX state space form (Nerrand, 1994,1993):

$$x\ (k)_p = f\ (x(k-1)\ , u\ (k-1)\ ) + e\ (k) \qquad (3.24)$$

where

$x_p\ (k) = \{\ x_1 , \ldots\ldots , x_n\ \}$ is a $n \times 1$ state vector .

$f(.)$ is a $n \times 1$ nonlinear vector function .

$u(k) = \{ u_1(k-1), \ldots, u_1(k-b), \ldots, u_m(k-1), \ldots u_m(k-b) \}$ is a $m \times b$ input matrix and $e(k)$ is the noise term ($n \times 1$ vector) .

For our multi input model in the above representation $n = 2$, $m = 2$, $b = 1$.

The regressors (or the input vector x_i to the network) will be :

$\varphi = \{x_1(k-1) , x_2(k-1) , x_3(k-1) , x_4(k-1) \}$.

where x_1 is equivalent to the state variable P_m (manifold pressure) and x_2 is equivalent to the state variable N and $x_3(k-1)$ is equivalent to the first input Th (the throttle command), $x_4(k-1)$ is equivalent to the second input D (The spark advance command).

3.5 Design of the Identification Experiment

Based on the previous discussion we need to design an identification experiment in which we handle the engine as a black box injected with two different input signals (throttle angle and spark timing) and producing two different output signals (engine speed and manifold pressure). Both of the input and the output signals are measured and through this data a mathematical dynamic model is constructed to emulate the engine dynamics.

Designing an identification experiment is a matter of experience and ad hoc, which differs from a plant to another. Among the most important issues to be considered are the input characteristics, the sampling interval of collecting the data and the amount of data points required. The input signal must be rich enough to excite the whole possible modes of interest over the whole amplitude range of operation of the plant which is the condition of "persistently exciting "of a signal (Johansson,1993). Also it

should be of different amplitudes (Ljung,1994) conforming with the practical constraints of the plant and give the plant a chance to more or less settle to give insight about the rise time of the plant.

Two practical sources of constraints affected our experiment. The first arises from the signals issuing device and the second arises from the plant itself. The first constraint source in our experiment is produced by the DC motor which actuates the throttle valve (in a step up-step down time steps) through a wire drive. That gives rise to the constraint that the input period allowed to the signal must not be less than 0.3-0.4 seconds, otherwise the input signal will be completely or partly distorted. Under this constraint the first input signal was chosen as a variable frequency, variable amplitude signal (a rough form of a pseudo random binary sequence signal).

The plant constraint comes from the fact that the plant responds very fast to small changes of the throttle. This means that the input signal must change in a manner allowing the dynamics of the engine to change over its whole amplitude range of operation to reflect how will be its behavior when rising and settling. According to this the input signal was made of small period in its maximum condition and large period in its minimum condition with variable amplitude and frequency in both conditions (a rough form of a pseudo random binary sequence signal) see Figure 3.3.

The second input signal which is the ignition timing is commanded to the plant through manifold pressure-spark timing map where we linked the signal value to manifold pressure variations through open ECU ignition map (a manifold pressure - ignition timing map). This signal also has the form of a rough PRBS signal and have

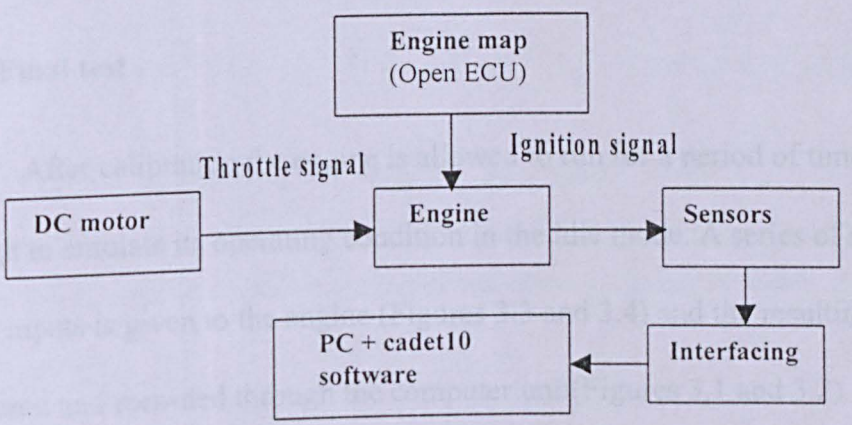
maximum - minimum range values which can practically be applied to the engine (in the interval [7 ,41]) see Figure 3.4.

Sampling interval is an important issue concerning collecting the data. It can be chosen to be correspondent to 5-8 sampling points over the rise time of the plant 's step response (Ljung,1994) and it is better to sample too fast than too slow ;also the aliasing effect must be considered. We used a sampling rate for collecting the data of a value equal to 100 points per second.

The amount of data required is a constraint imposed by the processing power of the computer used for applying the modeling algorithm computations. It is indirectly related to sampling interval where high sampling rate implies a redundant amount of data but this can be treated by decimating the data (Ljung, 1994).

3.6 Experiment Procedure

Experiment procedure is performed in two stages, the first one is the calibration stage and the second one is the test stage. Schematic diagram of the experiment is shown below(see also appendix A).



3.6.1 Engine calibration

The subject of the test experiment is a four cylinder spark ignition engine without any control attachments. The engine is connected to an open ECU without any control strategy installed in it. This implies that the engine should be first calibrated to run in its idle mode under the action of the throttle valve and ignition advance command from the open ECU. Through trial and error the roughly optimum control actions for the idle mode were 6 degrees for the throttle angle and 7.5 degrees before the top dead center for the spark timing. The engine under these conditions is running at slightly fluctuating speed around 750 r.p.m and manifold pressure around 450 kPa. Another important calibration was done to test the engine under disturbance that was necessary to estimate the maximum control action to be taken to stabilize the states of the engine under disturbance when running in its idle mode. This was found to be 25 degrees for the throttle angle and 41 degrees before top dead center for spark timing. This also was vital to the input magnitude design and the output magnitude range as the ANN models and the fuzzy controllers are some sort of autoassociative memory that can only interpolate.

3.6.2 Final test

After calibration the engine is allowed to run for a period of time to be warm enough to emulate its operating condition in the idle mode. A series of step up-step down inputs is given to the engine (Figures 3.3 and 3.4) and the resulting outputs are measured and recorded through the computer unit(Figures 3.1 and 3.2). A number of tests were run under the same input conditions to compensate for errors which may

result from the noise contaminating the readings of the measuring equipments and to compensate for variable operating conditions.

3.7 Results and Discussion of the Modeling Process

Preprocessing of the data involved removal of the data outliers only, the mean value or trends of the data were not removed as the constant term will compensate for them (Chen,1991).

A data record of *925* pairs was used to fit the model. In the first step a set of 925 regressors g_i is formed (including a constant term) with one delay unit for both the output variables N and P_m and the input variables T,D. A network of 925 nodes with Gaussian nonlinearity and initial value of $\zeta = 0.0003$ was initialized and the selecting procedures explained above were implemented in an iterative manner until the errors term in φ converge to the possible minimum value. The value of ζ is chosen guided by the discussion of section 2.5; in all cases the values of ζ less than 0.0016 led to over fitting. The nonlinearity in the hidden layer was taken as the Gaussian function form:

$$g_1 (X) = exp (X)^2 / 2\sigma^2$$

where $$\varphi = \left\| (x_1 - c_1)^2 + (x_2 - c_2)^2 + (x_3 - c_3)^2 + (x_4 - c_4)^2 + (x_5 - c_5)^2 + (x_6 - c_6)^2 \right\|_2 \qquad (3.25)$$

$x_1 = T(k-1)$, $x_2 = D(k-1)$, $x_3 = N(k-1)$, $x_4 = P(k-1)$

$x_5 = e_1(k)$ $x_6 = e_2(k)$ where e_1 is the speed error e_2 is the pressure error and $\| \|$ is the Euclidean norm.

Thus the regressors vector will be:

$$\varphi_r = [T(k\text{-}1)\ D(k\text{-}1)\ N(k\text{-}1)\ P(k\text{-}1)\ e_1(k)\ e_2(k)] \qquad (3.26)$$

After applying the OLS procedures to the RBF network the model was found to have the form:

$$y_p(k) = \alpha_o + \sum_{i=1}^{16} \hat{\theta}\, g(\ \|\ \varphi_r - C_i\ \|_2\ ) + e(k) \qquad (3.27)$$

where $\hat{\theta}$ is outer layer parameters (a 16×2 matrix) .

The set of C_i is given by $C=[c|c_1|c_2]$

$$c = \begin{bmatrix} 0.9869 & 0.9657 & 0.8970 & 0.9275 & 0.9900 & 0.9793 & 0.2684 \\ 0.9944 & 0.9931 & 0.9972 & 0.9924 & 0.9904 & 0.9933 & 0.1697 \\ 0.8494 & 0.9724 & 0.3606 & 0.2633 & 0.4445 & 0.4372 & 0.9894 \\ 0.7208 & 0.6384 & 0.4995 & 0.6533 & 0.9144 & 0.8134 & 0.2831 \end{bmatrix}$$

$$c_1 = \begin{bmatrix} 0.3524 & 0.9680 & 0.9852 & 0.3673 & 0.3306 & 0.3670 & 0.3689 \\ 0.1653 & 0.9940 & 0.9986 & 0.1739 & 0.1689 & 0.9958 & 0.9904 \\ 0.4351 & 0.3045 & 0.5243 & 0.2417 & 0.4029 & 0.9384 & 0.2933 \\ 0.2551 & 0.9624 & 0.9013 & 0.4321 & 0.4065 & 0.5926 & 0.5399 \end{bmatrix}$$

$$c_2 = \begin{bmatrix} 0.2778 & 0.3388 \\ 0.1742 & 0.1720 \\ 0.9408 & 0.2861 \\ 0.3403 & 0.3508 \end{bmatrix}$$

The matrix $\hat{\Gamma}$ is given by:

$$\hat{\Gamma} = \begin{bmatrix} 0.5129 & 0.5049 \\ 14.7949 & -18.1493 \\ -2.9547 & -2.4148 \\ 9.3792 & 26.3677 \\ 27.9733 & -31.7979 \\ 37.9813 & 95.6234 \\ 0.2170 & -0.0074 \\ -3.1684 & -1.8885 \\ -37.7862 & 70.8630 \\ -12.9692 & -138.48 \\ -1.2844 & 8.0316 \\ -22.2706 & -200.99 \\ -3.5719 & 8.4571 \\ 4.4615 & -24.1425 \\ -11.5229 & -22.6919 \\ -90.6233 & -430.3944 \end{bmatrix}$$

The matrix B is given by:

$$B = \begin{bmatrix} 1 & 0.9913 & 1.0627 & 1.046 & 1.0059 & 1.0224 & 1.1165 & 1.1805 & 0.9995 & 1.0055 & 1.1793 & 1.1846 & 1.0599 & 1.1116 & 1.1288 & 1.1818 \\ 0 & 1 & -0.3184 & -1.6378 & -2.5666 & -2.0489 & 10.7295 & 8.7200 & -3.2509 & -2.227 & 6.781 & 7.8253 & 3.6255 & 1.4324 & 10.2710 & 7.5604 \\ 0 & 0 & 1 & 0.8659 & 0.2255 & 0.371 & 3.342 & 4.046 & 0.3542 & 0.1456 & 4.0306 & 3.9706 & 1.2344 & 2.1049 & 3.3515 & 4.1264 \\ 0 & 0 & 0 & 1 & 1.5828 & 0.9904 & 35.0403 & 34.1998 & 2.4514 & 1.2266 & 36.186 & 36.7372 & 11.0655 & 10.7130 & 35.7519 & 35.5538 \\ 0 & 0 & 0 & 0 & 1 & 0.5725 & -6.47 & -16.6875 & 1.6786 & 1.1988 & -20.1388 & -17.3729 & 10.641 & 6.9683 & -7.6295 & -18.999 \\ 0 & 0 & 0 & 0 & 0 & 1 & -102.9242 & -134.542 & -0.1999 & -0.1839 & -146.1015 & -141.4062 & -33.5204 & -44.6232 & -107.6492 & -145.35 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1.0659 & -0.0115 & -0.0109 & 1.0234 & 1.0386 & -0.1108 & -0.111 & 0.9976 & 1.0366 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -0.0185 & -0.0276 & 0.982 & 0.7477 & -0.9919 & -0.8724 & -0.04 & 1.051 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1.3796 & 21.133 & 20.2994 & 57.599 & 51.9496 & 8.4173 & 12.383 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 17.1064 & 15.137 & 52.0206 & 30.3104 & 7.292 & 8.892 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8704 & 1.048 & 1.189 & 0.4249 & 0.385 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 4.6746 & 3.077 & 0.9913 & -0.0874 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1.001 & -0.0695 & 0.114 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -0.091 & 0.1123 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1.738 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The set of the outer linear parameters is given by

$$\hat{\theta} = \begin{bmatrix} -0.8862 & -58.6023 \\ 8.3020 & 29.5961 \\ -0.0357 & -32.1639 \\ 21.7317 & 84.9413 \\ 164.5250 & -131.76 \\ -80.8180 & -59.5667 \\ 42.9980 & 10.3037 \\ -66.2205 & 164.2967 \\ -41.1091 & -23.2040 \\ -70.9367 & 192.7255 \\ -110.107 & 310.0413 \\ 35.7035 & 12.1405 \\ -7.1701 & -3.7111 \\ -2.8292 & -7.9520 \\ -46.9839 & -23.6342 \\ 157.7728 & -511.83 \end{bmatrix}$$

where $\hat{\theta} = B^{-1}\hat{\Gamma}$ .The final model form is:

$$y_{p1}(k) = A(\alpha_{o1} + \sum_{i=1}^{16} \exp\left\{\left[(aT(k-1)-c_{1i})^2 + (bD(k-1)-c_{2i})^2 + (cN(k-1)-c_{3i})^2 + (dP(k-1)-c_{4i})^2 + (e_1(k)-c_{5i})^2 + (e_2(k)-c_{6i})^2\right] / \sigma^2\right\})$$
$$+e_1(k)) \qquad (3.28)$$

$$y_{p2}(k) = B(\alpha_{o2} + \sum_{i=1}^{16} \exp\left\{\left[(aT(k-1)-c_{1i})^2 + (bD(k-1)-c_{2i})^2 + (cN(k-1)-c_{3i})^2 + (dP(k-1)-c_{4i})^2 + (e_1(k)-c_{5i})^2 + (e_2(k)-c_{6i})^2\right] / \sigma^2\right\})$$
$$+e_2(k)) \qquad (3.29)$$

A=2426.9 *B=97.28*

where

$$\hat{\theta}_1 = \begin{bmatrix} -0.8862 \\ 8.3020 \\ -0.0357 \\ 21.7317 \\ 164.525 \\ -80.818 \\ 42.9980 \\ -66.2205 \\ -41.1091 \\ -70.9367 \\ -110.107 \\ 35.7035 \\ -7.1701 \\ -2.8292 \\ -46.9839 \\ 157.7728 \end{bmatrix}$$

$$\hat{\theta}_2 = \begin{bmatrix} -58.6023 \\ 29.5961 \\ -32.1639 \\ 84.9413 \\ -131.7587 \\ -59.5667 \\ 10.3037 \\ 164.2967 \\ -23.2040 \\ 192.7255 \\ 310.0413 \\ 12.1405 \\ -3.7111 \\ -7.9520 \\ -23.6342 \\ -511.8278 \end{bmatrix}$$

a=0.0386 *b=0.024* *c= 0.0004122* *d=0.103*

The corresponding state space form will be:

$$x_{p1}(k) = A(-0.0011 + \sum_{i=1}^{16} \exp(\|(au_1(k-1)-c_{1i})^2 + (bu_2(k-1)-c_{2i})^2 + (cx_1(k-1)-c_{3i})^2 + (dx_2(k-1)-c_{4i})^2 + (e_1(k)-c_{5i})^2 + (e_2(k)-c_{6i})^2\| / \sigma^2)) + e_1(k)$$

(3.30)

$$x_{p2}(k) = B(-0.14 + \sum_{i=1}^{16} \exp(\|(au_1(k-1)-c_{1i})^2 + (bu_2(k-1)-c_{2i})^2 + (cx_1(k-1)-c_{3i})^2 + (dx_2(k-1)-c_{4i})^2 + (e_1(k)-c_{5i})^2 + (e_2(k)-c_{6i})^2\| / \sigma^2)) + e_2(k)$$

(3-31)

The deterministic forms of the model are given by:

$$y_{p1}(k) = A(-0.0011 + \sum_{i=1}^{16} \exp(\|(aT(k-1)-c_{1i})^2 + (bD(k-1)-c_{2i})^2 + (cN(k-1)-c_{3i})^2 + (dP(k-1)-c_{4i})^2\| / \sigma^2))$$

(3.31)

$$y_{p2}(k) = B(-0.14 + \sum_{i=1}^{16} \exp(\|(aT(k-1)-c_{1i})^2 + (bD(k-1)-c_{2i})^2 + (cN(k-1)-c_{3i})^2 + (dP(k-1)-c_{4i})^2\| / \sigma^2))$$

(3.32)

$$x_{p1}(k) = A(-0.0011 + \sum_{i=1}^{16} \exp(\|(au_1(k-1)-c_{1i})^2 + (bu_2(k-1)-c_{2i})^2 + (cx_1(k-1)-c_{3i})^2 + (dx_2(k-1)-c_{4i})^2\| / \sigma^2))$$

(3.33)

$$x_{p2}(k) = B(-0.14 + \sum_{i=1}^{16} \exp(\|(au_1(k-1)-c_{1i})^2 + (bu_2(k-1)-c_{2i})^2 + (cx_1(k-1)-c_{3i})^2 + (dx_2(k-1)-c_{4i})^2\| / \sigma^2))$$

(3.34)

$\alpha_{o1} = -0.0011$ $\alpha_{o2} = -0.14$

The fitting and prediction results of the model are given in Figures 3.1 to 3.10 at the end of the chapter.

An important factor which is important to consider, is the threshold value for the product $w^T w$, where this value affect the number of vectors involved in the orthogonalization process. Too small value of the threshold increases the number of vectors involved in the process leading to spurious orthogonalized vectors. The selected auxiliary regressors (or subset regressors) were found to be sixteen regressors which implies that the number of the basis functions is equal to sixteen and the number of the auxiliary parameters is thirty two parameters (sixteen for each output). The final network structure is sixteen nodes in the hidden layer, six input nodes and two output nodes (see Figure 3.11). In the beginning of the modeling process different model structures with different orders were tried, namely the ARX (or the equation error) and the NARMAX structures.

The ARX requires a directed algorithm in its training, while the NARMAX requires a semi-directed algorithm (Nerrand, 1994,1993). In training both the structures the NARMAX one could efficiently fit the model better than the ARX structure. Four iterations only were enough for the NARMAX structure to fit the model and yield the minimum error. Also different model orders were tried but the results were not superior to the first order assumption. Verification of the first order assumption is checked for in the next chapter, where it can be concluded that the resulting model is giving good results and able to reproduce the plant dynamics. Fittting accuracy can be seen from Figures 3.1,3.2 and 3.5,3.6 where the input vector is given by $\varphi = [u_1(k-1)\ u_2(k-1)\ y_1(k-1)\ y_2(k-1)\ e_1(k-1)\ e_2(k-1)]$, where $y_1(k-1)$ and $y_2(k-1)$ are the real plant outputs. Prediction accuracy check of the model is performed by the input vector: $\varphi = [u_1(k-1)\ u_2(k-1)\ \hat{y}_1(k-1)\ \hat{y}_2(k-1)]$, where $\hat{y}_1(k-1)$ and $\hat{y}_2(k-1)$ are

the estimated model outputs. The prediction accuracy of the model can be seen in Figures 3.9 and 3.10 which is good for model derived from a real plant. Adequacy of the model is checked for in the next chapter where the first order assumption is justified. In the experiment the plant was excited by a not so random signal; exciting the system with a real random signal could be better to represent more efficiently the dynamics of the system, but this couldn't be done because of the hardware constraints of the signal issuing devices.

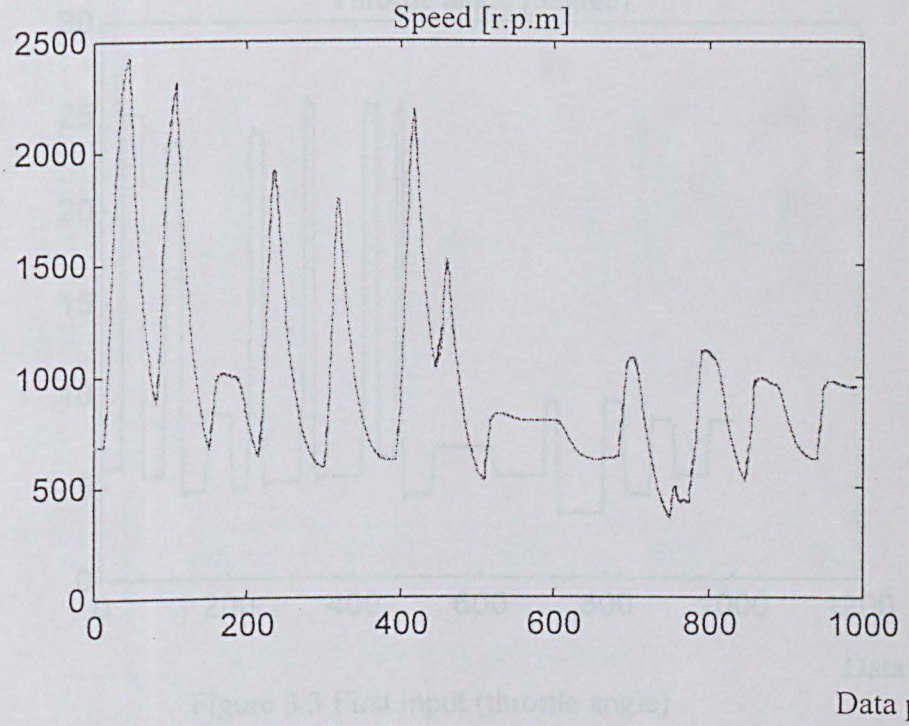


Figure 3.1 Speed (— model --- plant)

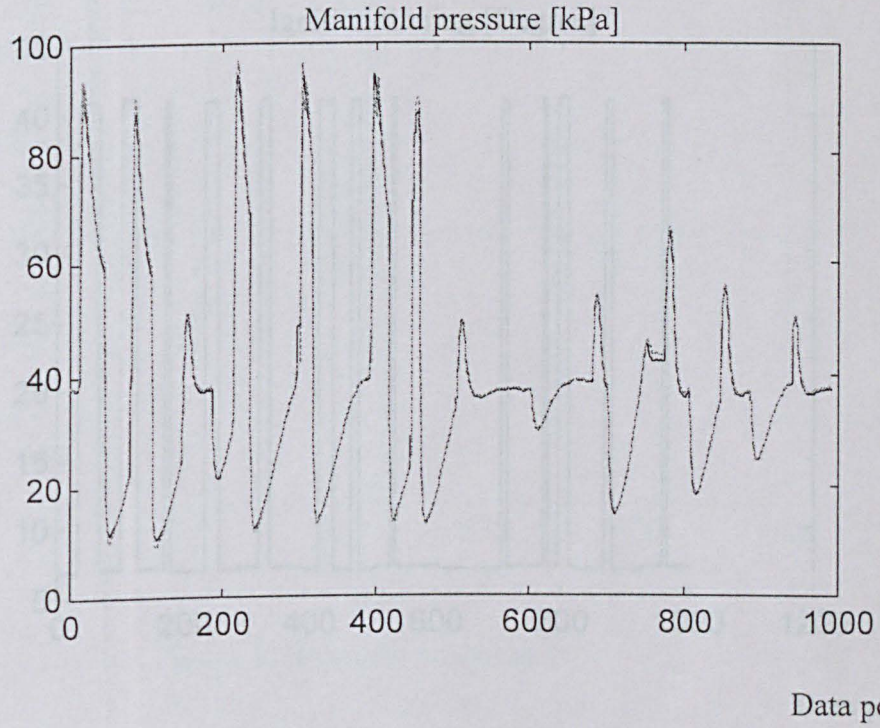


Figure 3.2 Manifold pressure (— model --- plant)

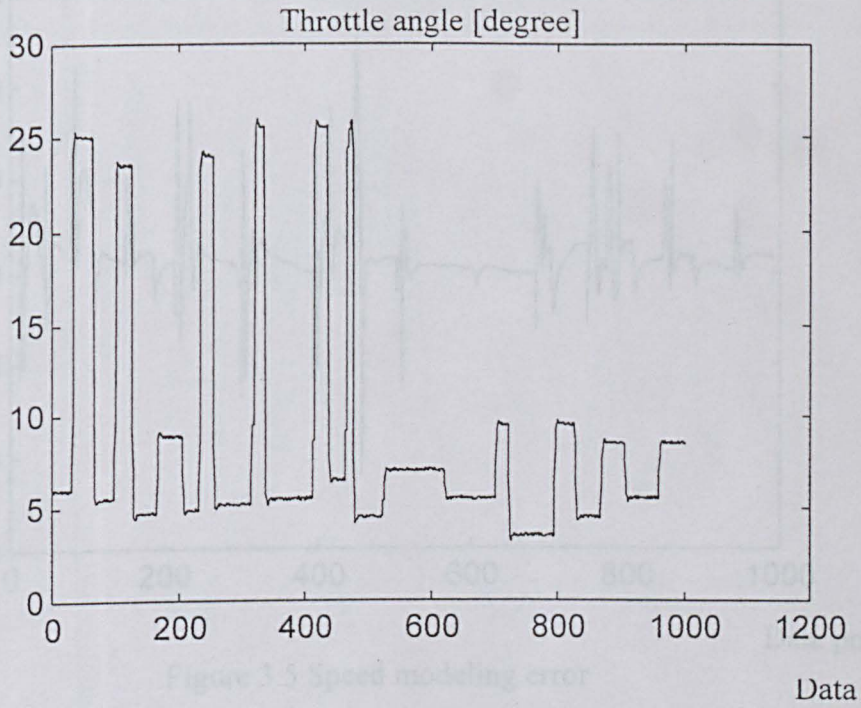


Figure 3.3 First input (throttle angle)

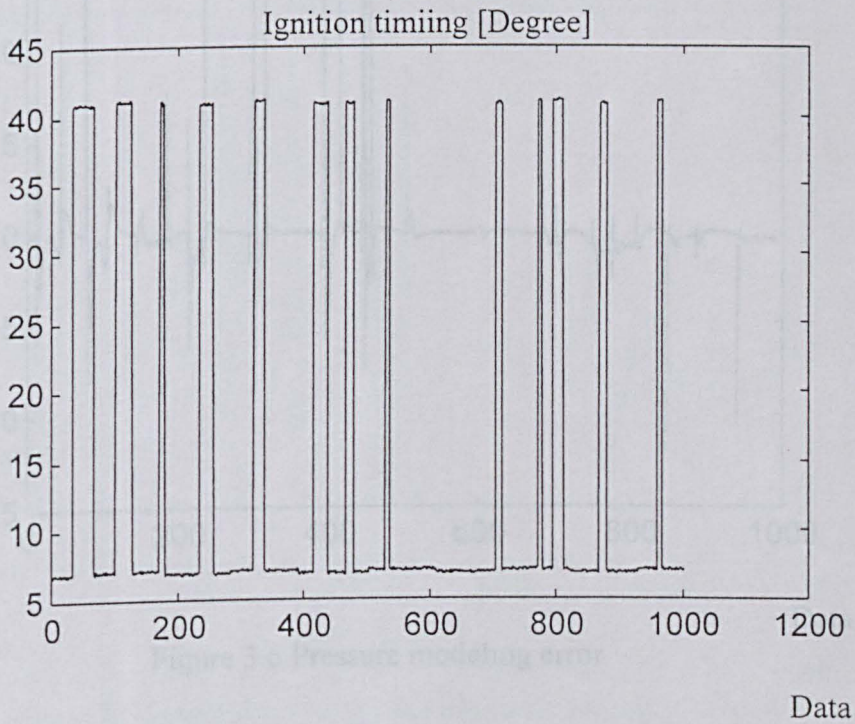


Figure 3.4 Second Input (Ignition timing)

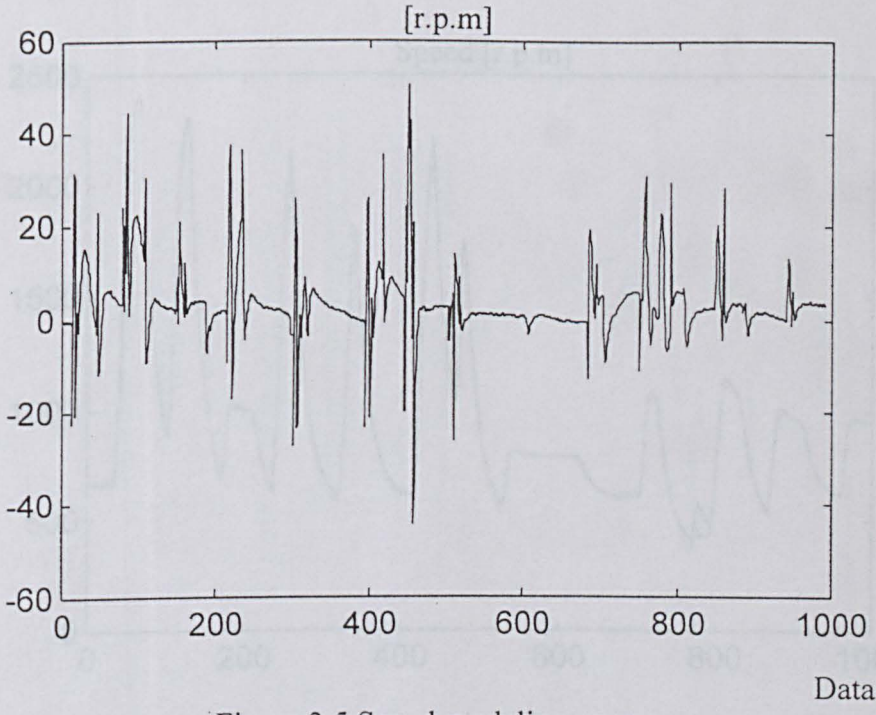


Figure 3.5 Speed modeling error

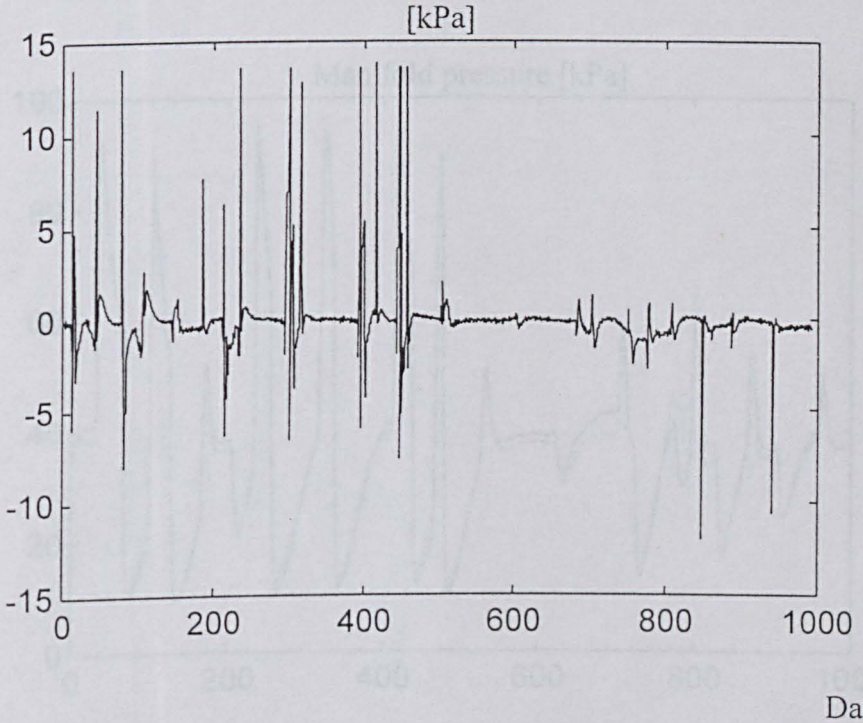


Figure 3.6 Pressure modeling error

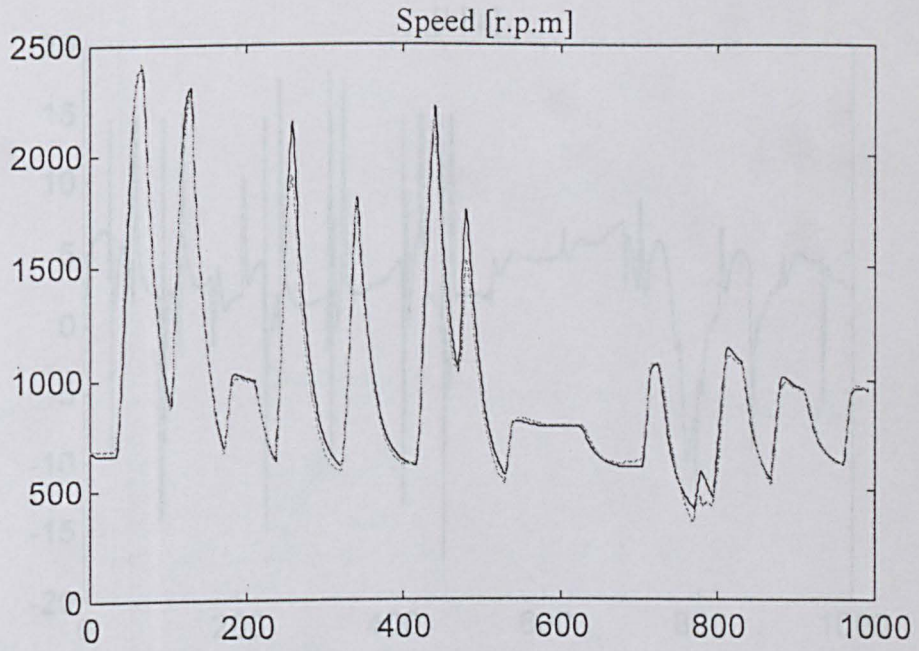


Figure 3.7 Speed deterministic output (— model --- plant)

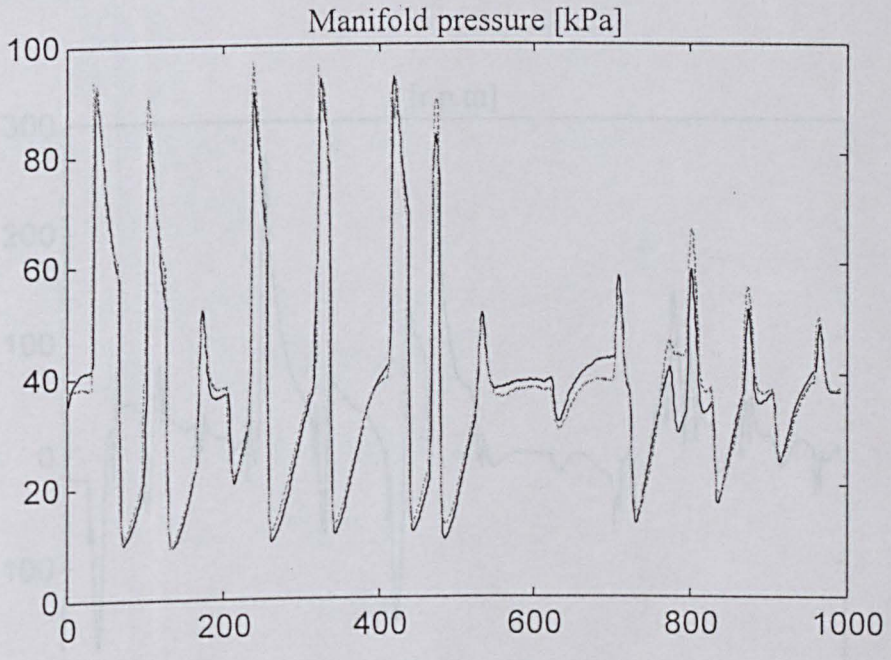


Figure 3.8 Manifold pressure deterministic output (— model --- plant)

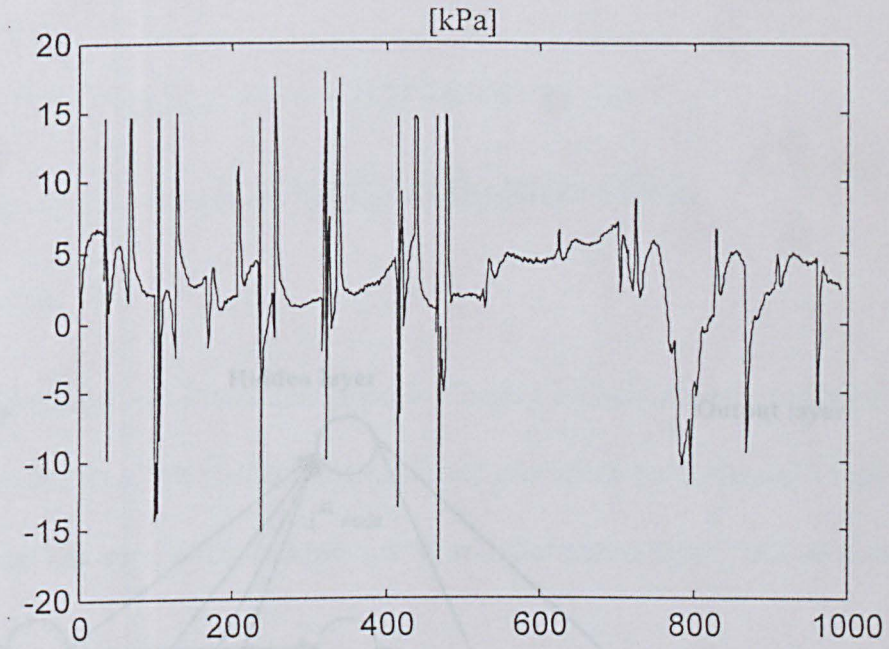


Figure 3.9 Pressure deterministic error

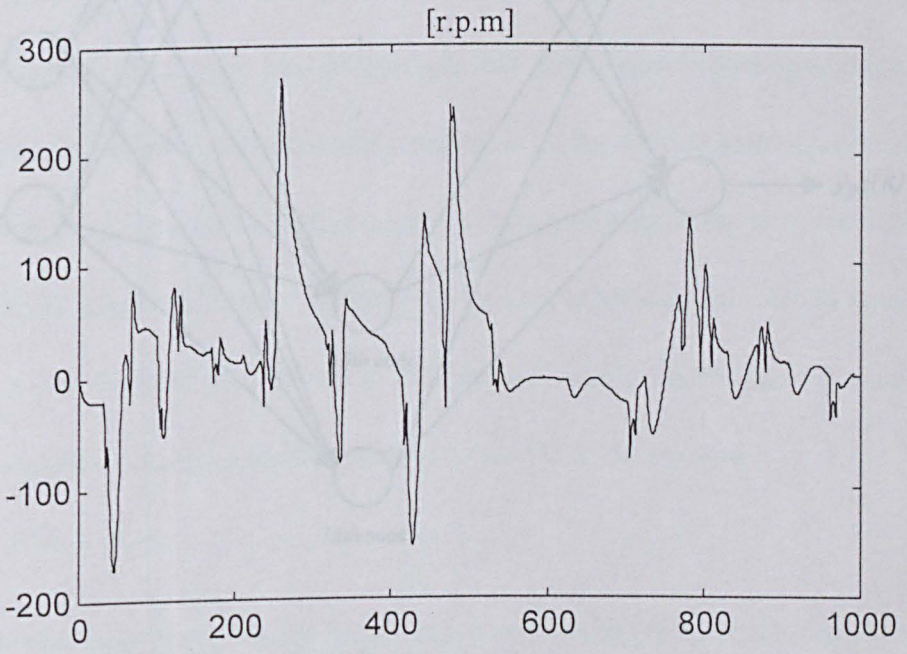


Figure 3.10 Speed deterministic error

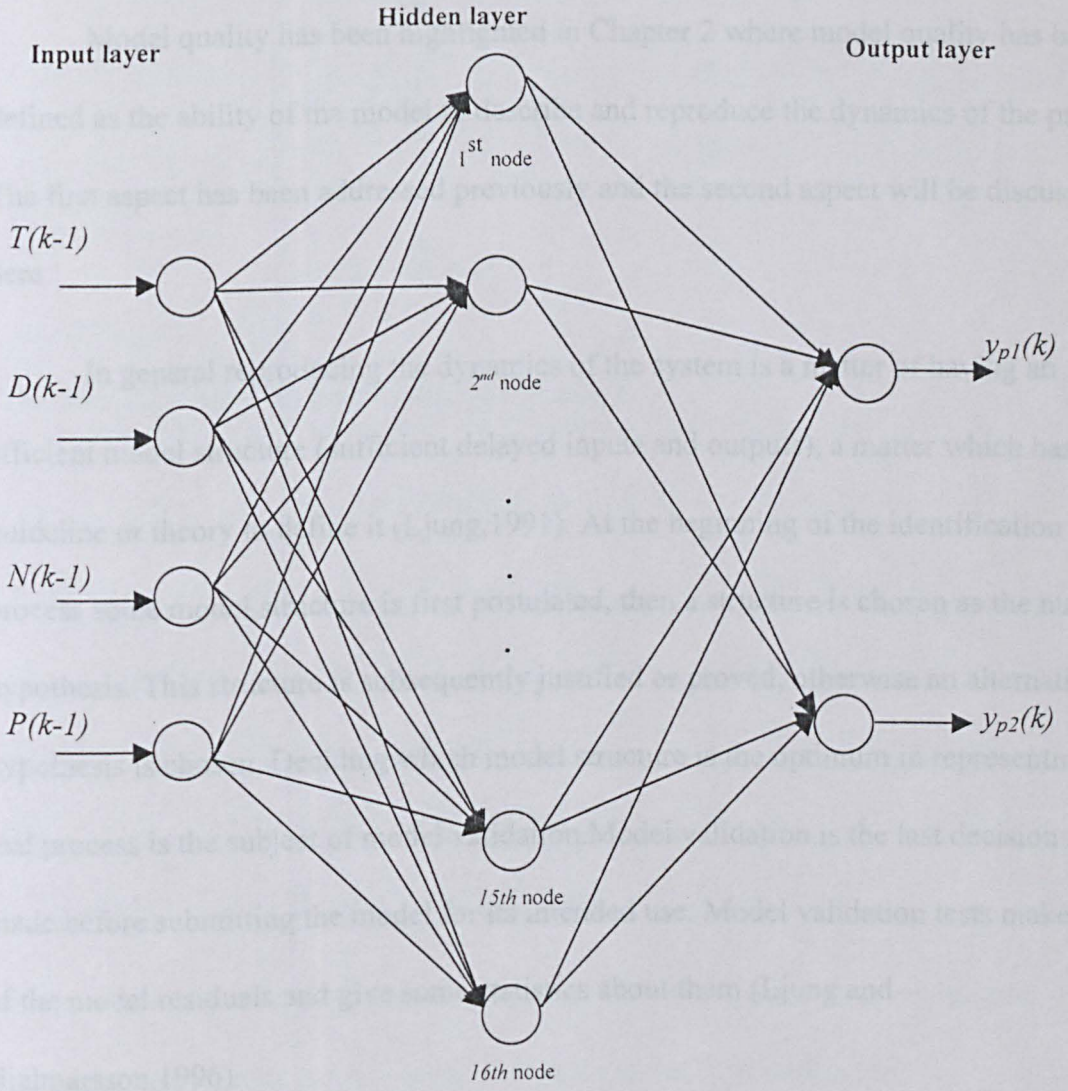


Figure 3.11 Final network structure

CHAPTER 4

MODEL VALIDATION TESTS

4.1 Introduction

Model quality has been highlighted in Chapter 2 where model quality has been defined as the ability of the model to describe and reproduce the dynamics of the process. The first aspect has been addressed previously and the second aspect will be discussed here.

In general reproducing the dynamics of the system is a matter of having an efficient model structure (sufficient delayed inputs and outputs), a matter which has no guideline or theory to define it (Ljung,1991). At the beginning of the identification process some model structure is first postulated, then a structure is chosen as the null hypothesis. This structure is subsequently justified or proved, otherwise an alternative hypothesis is chosen. Deciding which model structure is the optimum in representing the real process is the subject of model validation.Model validation is the last decision to be made before submitting the model for its intended use. Model validation tests make use of the model residuals and give some statistics about them (Ljung and Hjalmarsson,1996).

Model residuals are given by $\varepsilon = y - \hat{y}$ where y is the real process output and $\hat{y}$ is the identified model output. The statistics which are often used include:

(1) Maximum absolute value of the residuals

$$[\varepsilon]_{max} = max\ |\varepsilon(t)| \qquad 1 \le t \le N \tag{4.1}$$

(2) Mean Variance and Mean square of the residuals

$$\mu_{\varepsilon} = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t) \tag{4.2}$$

$$V_{\varepsilon} = \frac{1}{N}\sum_{t=1}^{N}(\varepsilon(t)-\mu)^2 \tag{4.3}$$

$$\mu'_{\varepsilon} = \frac{1}{N}\sum_{t=1}^{N}\varepsilon(t)^2 \tag{4.4}$$

(3) Correlation between residuals and past inputs

$$\phi_{u\varepsilon} = E\{u_j\varepsilon_i\} = o \quad \forall\ i, j \tag{4.5}$$

Several methods exist for model validation [Leontarits and Billings (1987), Johansson (1993)]. Some of these methods rely on comparing postulated different models of different orders and parameters and then based on some statistical criterion a structure is chosen. The F-test, Akiake information criterion (AIC) and Akiake final prediction error (FPE) are examples of these methods. Other methods rely on analyzing the residuals and its whiteness, normality and independence of both lagged inputs and outputs. These are nonparametric tests and an example of them is the 'correlation tests'.

Model validation problem can be formulated as a statistical hypothesis testing problem as we have to decide which model can best represent the real process. A null hypothesis H_O

and an alternative hypothesis H_1 is postulated. The null hypothesis is the one that is not rejected unless the data provide strong evidence that it is not correct. It is always taken as the identified model.

In this study the nonlinear correlation tests developed by Billing and his coworkers [Billings and Voon (1983), Billings and Voon (1986), Leontaritis and Billing, (1987) Billings and Zhu (1994, Billings and Zhu (1995)] will be used for the purpose of model validation.

4.2 **Correlation Analysis**

Mathematically correlation of two random variables is equal to the time average of the product of their expected values; it indicates to which extent the variables are dependent on each other. If the variables come from the same signal, the correlation function between the signals at two different time instants is called autocorrelation function which is a measure of the predictability of the signal at the future time based on knowledge of the present value of the signal. When the variables come from two different signals the correlation function is called the crosscorrelation function.

The autocorrelation function is defined as

$$\varphi_{\varepsilon\varepsilon} = E\{\varepsilon(t)\varepsilon(t+\tau)\} = \lim \frac{1}{2T} \int_{-T}^{T} \varepsilon(t)\varepsilon(t+\tau)dt \tag{4.6}$$

The crosscorrelation function is defined as

$$\varphi_{u\varepsilon} = E\{u(t)\varepsilon(t+\tau)\} = \lim \frac{1}{2T} \int_{-T}^{T} u(t)\varepsilon(t+\tau)dt \tag{4.7}$$

In model validation correlational tests usually amount to calculating the autocorrelation function of the residuals $\varphi_{\varepsilon\varepsilon}$ and the crosscorrelation function $\varphi_{u\varepsilon}$ between the residuals and the input. Then to check if both of them are asymptotically normal with zero mean and finite variance or not, where the standard deviations are $\frac{1}{\sqrt{N}}$ and the 95% confidence levels are approximately $\frac{1.96}{\sqrt{N}}$. Traces of any lagged inputs or outputs in the residuals can be discovered if $\varphi_{u\varepsilon}$ happens to be outside the confidence levels while traces of lagged residuals (dependent or future predicted residuals) are checked for by confidence levels banded $\varphi_{\varepsilon\varepsilon}$. This is given by:

$$\varphi_{\varepsilon\varepsilon}(\tau) = \begin{cases} 1, & \tau = 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.8}$$

$$\varphi_{u\varepsilon} = 0, \qquad \forall \tau \qquad (\tau = \text{time steps})$$

Correlation model validity tests are straightforward for linear systems. They are given in any of Billing et al previous papers. Linear correlation tests fail to diagnose inadequacies in the nonlinear models. A number of tests for nonlinear models are given in the previous references. The tests given in Billings and Zhu (1995) are used here. As stated in Billings and Zhu (1994,1995) the involvement of the output enhances the discriminatory performance compared with the tests based only on the input and residuals. The test equations are given as follows:

$$\varphi_{\zeta\eta} = E\{\zeta(t)\eta(t+\tau)\} \tag{4.9}$$

$$\varphi_{\vartheta\eta} = E\{\vartheta(t)\eta(t+\tau)\} \tag{4.10}$$

where

$$\zeta(t) = \varepsilon^2_1(t) + \ldots\ldots + \varepsilon^2_q(t) \tag{4.11}$$

$$\eta(t) = y_1(t)\varepsilon_1(t) + \ldots\ldots + y_q(t)\varepsilon_q(t) \tag{4.12}$$

$$\vartheta(t) = u^2_1(t) + \ldots\ldots + u^2_r(t) \tag{4.13}$$

The normalized correlation tests are

$$\varphi_{\zeta\eta}(\tau) = \frac{\sum_{t=1}^{N-\tau} \zeta^\circ(t)\eta^\circ(t-\tau)}{\left(\sum_{t=1}^{N}(\zeta^\circ(t))^2 \sum_{t=1}^{N}(\eta^\circ(t))^2\right)^{1/2}} \tag{4.14}$$

$$\varphi_{\eta}(\tau) = \frac{\sum_{t=1}^{N-\tau} \vartheta^\circ(t)\eta^\circ(t-\tau)}{\left(\sum_{t=1}^{N}(\vartheta^\circ(t))^2 \sum_{t=1}^{N}(\eta^\circ(t))^2\right)^{1/2}} \tag{4.15}$$

where

$$\zeta^\circ(t) = \frac{\varepsilon^{2\circ}_1(t)}{\left(\frac{1}{N}\sum_{1}^{N}(\varepsilon^{2\circ}_1(t))^2\right)^{1/2}} + \ldots\ldots + \frac{\varepsilon^{2\circ}_q(t)}{\left(\frac{1}{N}\sum_{1}^{N}(\varepsilon^{2\circ}_q(t))^2\right)^{1/2}} \tag{4.16}$$

$$\eta^\circ(t) = \frac{(y_1(t)\varepsilon_1(t))^\circ}{\left(\frac{1}{N}\sum_{1}^{N}(y_i(t)\varepsilon_i(t))\right)^{1/2}} + \ldots\ldots + \frac{(y_q(t)\varepsilon_q(t))^\circ}{\left(\frac{1}{N}\sum_{1}^{N}(y_i(t)\varepsilon_i(t))\right)^{1/2}} \tag{4.17}$$

$$\vartheta^\circ(t) = \frac{u^{2\circ}_1(t)}{\left(\frac{1}{N}\sum_{1}^{N}(u^{2\circ}_1(t))^2\right)^{1/2}} + \ldots\ldots + \frac{u^{2\circ}_r(t)}{\left(\frac{1}{N}\sum_{1}^{N}(u^{2\circ}_r(t))^2\right)^{1/2}} \tag{4.18}$$

$$\varepsilon_i^{2\circ}(t) = \varepsilon_i^2(t) - \overline{\varepsilon_i^2} \qquad \overline{\varepsilon_i^2} = \frac{1}{N}\sum_{t=1}^{N} \varepsilon_i^2(t) \tag{4.19}$$

$$(y_i(t)\varepsilon_i(t))^\circ = y_i(t)\varepsilon_i(t) - \overline{y_i\varepsilon_i} \qquad \overline{y_i\varepsilon_i} = \frac{1}{N}\sum_{t=1}^{N} y_i\varepsilon_i(t) \tag{4.20}$$

$$u_j^{2\circ} = u_j^2(t) - \overline{u_j^2} \qquad \overline{u_j^2} = \frac{1}{N}\sum_{t=1}^{N} u_j^2(t) \tag{4.21}$$

$i=1 \ldots q, \quad j = 1 \ldots \ldots r$

The above tests check for correlation among all the sub-modes input, output and residual vectors. If every subsystem in the model is valid ($\varphi_{\varepsilon\varepsilon} = 0$) correlation tests will yield:

$$\varphi_{\zeta\eta}(\tau) = \begin{cases} k, & \tau = 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.22}$$

$$\varphi_{\vartheta\eta}(\tau) = 0, \ \forall \tau \tag{4.22}$$

$$k = \frac{\left(\sum_{t=1}^{N} (\zeta^\circ(t))^2\right)^{1/2}}{\left(\sum_{t=1}^{N} (\eta^\circ(t))\right)^{1/2}} \tag{4.24}$$

k is a constant.

If it happens to discover model global inadequacies, another set of local tests are used to specify the submodel source of these inadequacies.

4.3 Formulation of Model Validity Tests

Typical model validity statistical tests consist of the following three steps [Bohlin (1978)]:

(1) The first step is to find a parameter free statistics (say ε) which is a function of the validation data such that the distribution of ε is known if the null hypothesis H_O is true.

(2) The second step is to define a domain D^α such that

$$P\{\ \varepsilon \notin D^\alpha\ \ |H_o\} = \alpha$$

where α is the risk level of rejecting a model when it is actually valid.

(3) The third step is to reject H_O if $\varepsilon \notin D^\alpha$.

In residual analysis the formulation of the test problem is as follows

The null hypothesis is

$\{\varepsilon_k\}$ Comprise a white noise with zero mean and minimum possible variance

i.e. $E\{\varepsilon(t)\varepsilon(t+\tau)\} = 0.$

$\{\varepsilon_k\}$ Are normally and symmetrically distributed.

$\{\varepsilon_k\}$ Are independent (uncorrelated) of both lagged inputs and outputs i.e.

$E\{u(t)\varepsilon(t+\tau)\} = 0.$

The parameter free statistic will be the residuals vector ε.

The domain D^{α} is $D^{\alpha} = \{\varepsilon | \phi < K_{\alpha}\}$, ϕ is the test, K_{α} is the decision value and α is the probability of accepting the incorrect model $\alpha = 0.05$

The 95% confidence level $K_{\alpha} = K_{0.05} = \pm \frac{1.96}{\sqrt{N}}$ is used when ϕ is normally distributed. The hypothesis H_O is rejected when $\varepsilon \notin D^{\alpha}$.

4.4 Application of Correlation Tests to the Engine Idle Speed Nonlinear Model

The previously given tests (Equations 4.9 to 4.20) are applied to the MIMO nonlinear deterministic model:

$$y_{p1}(k) = A\left(\sum_{i=1}^{16} \hat{\theta}_1 \exp\left(\left\|(aT(k-1)-c_{1i})^2 + (bD(k-i)-c_{2i})^2 + (cN(k-1)-c_{3i})^2 + (dP(k-1)-c_{4i})^2\right\|_2\right) - 0.0011\right) \quad (4.25)$$

$$y_{p2}(k) = B\left(\sum_{i=1}^{16} \hat{\theta}_2 \exp\left(\left\|(aT(k-1)-c_{1i})^2 + (bD(k-i)-c_{2i})^2 + (cN(k-1)-c_{3i})^2 + (dP(k-1)-c_{4i})^2\right\|_2\right) + 0.14\right) \quad (4.26)$$

$y_1(k) = N(k)$ (speed) $y_2(k) = P(k)$ (manifold pressure) $(q=2)$ where model parameters are give in Chapter3.

First the model is tested with a data set (validation data set) different from that set used for fitting the model in chapter3, a data record of 1000 data points is used. Testing results are shown in Figures 4.1 to 4.8 After that the tests of Equations 4.3 to 4.4 were applied to the simulation results where:

The inputs are:

$u_1 = T(k-1)$ (throttle) $u_2 = D(k-1)$ (ignition timing) $(r=2)$

The outputs are:

$y_1(k) = N(k)$ (speed) $y_2(k) = P(k)$ (manifold pressure) $(q=2)$

The model residuals are:

$\varepsilon_1 = e_1(k)$ (speed error)

$\varepsilon_2 = e_2(k)$ (manifold pressure error)

$K=0.95$

4.5 Discussion and Conclusion

Testing of the model in its deterministic form is done with the validation data set. The regressor vector is composed of the delayed input and the delayed output of the model. It takes the form:

$$\varphi = [u_1(k-1)\ u_2(k-1)\ \hat{y}_1(k-1)\ \hat{y}_2(k-1)]$$

where, $\hat{y}_1(k-1)$, $\hat{y}_2(k-1)$ are the outputs of the estimated model.

It can be seen from Figures 4.1 and 4.2 that the model and the plant are very close to each other. The model is able to reproduce the plant dynamics efficiently for different input signals. This can be easily noticed from the difference between the input signal used for fitting the model and the input signal for validating the model in the last chapter, where the frequencies and the amplitudes of the input signals are different as can be seen

from Figures 4.3 and 4.4. Prediction errors of both the two outputs can be seen in Figures 4.5 and 4.6.

Correlation results are shown in Figures 4.7 and 4.8 where it can be seen that the values of the correlation functions $\varphi_{\zeta\eta}(\tau)$, $\varphi_{\vartheta\eta}(\tau)$ are within the bands of the confidence interval [0.05 -0.05]. This means that the model is able to reproduce the dynamics of the system for different input frequencies and amplitudes to the plant. This also implies that there is no need to increase the model order, as the first order for each submodel is able to reproduce the plant dynamics. Another implication is that the input signal was rich enough to excite the different frequency modes of the plant.

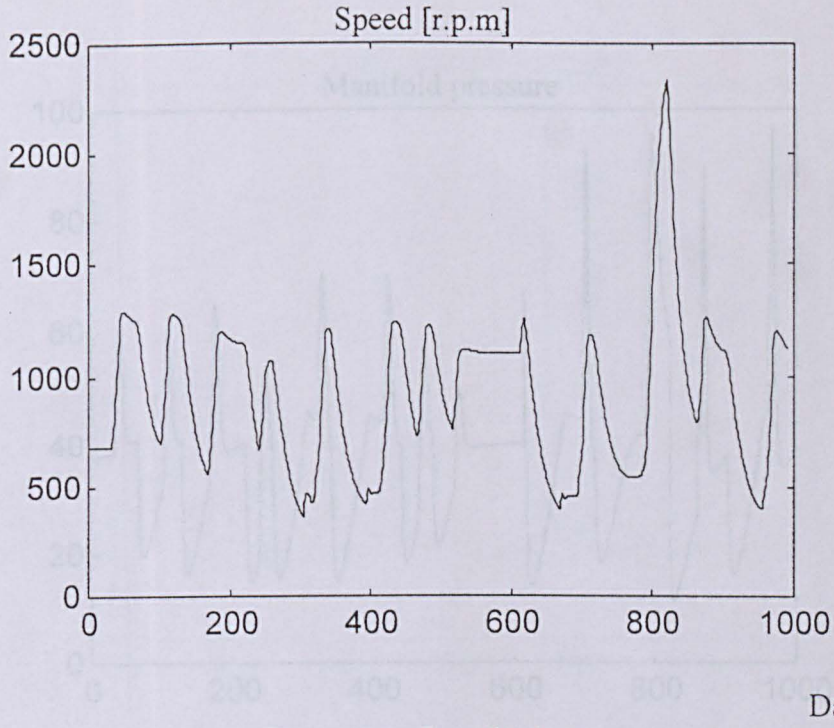


Figure 4.1a Speed (plant)

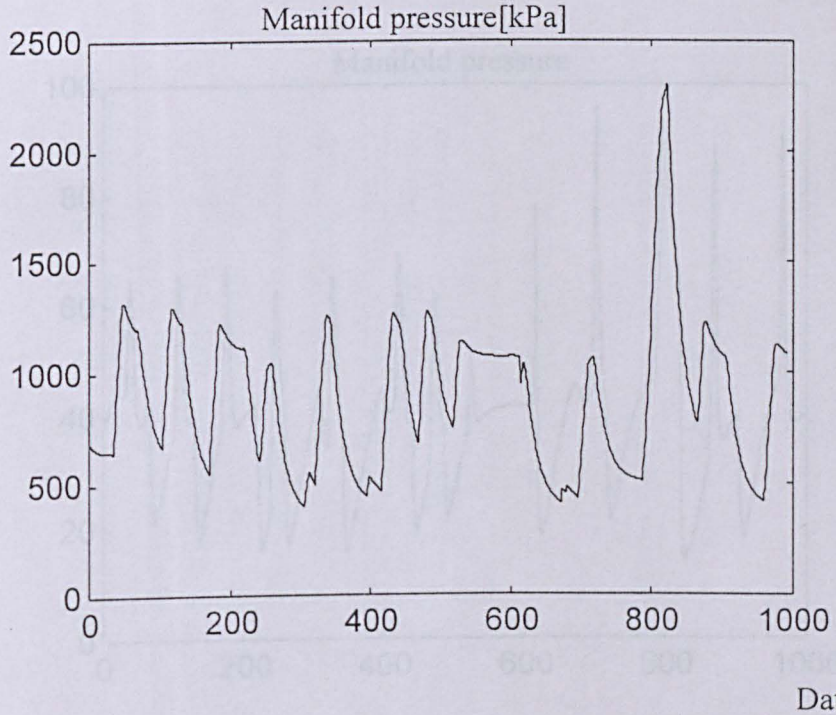


Figure 4.1b Speed (estimated model)

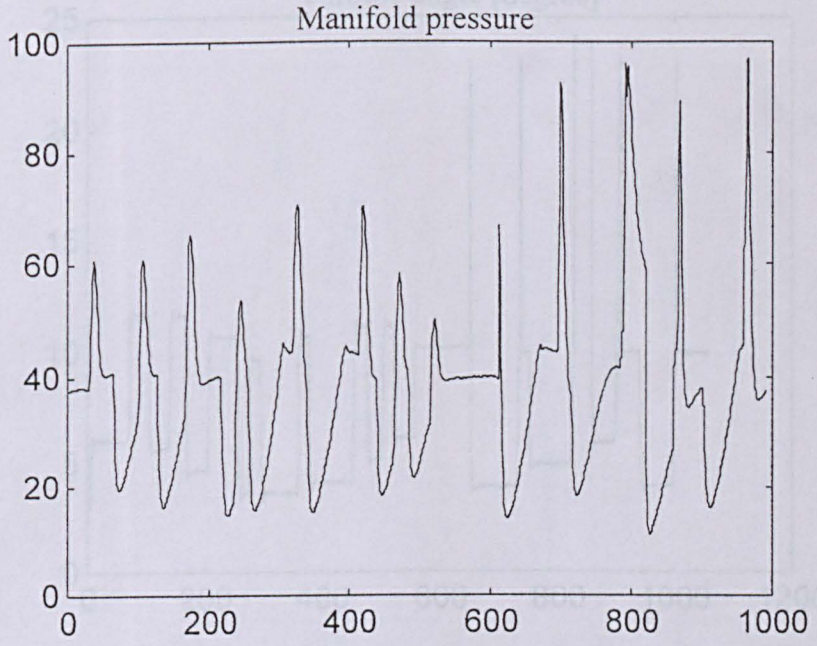


Figure 4.2a Manifold pressure (plant)

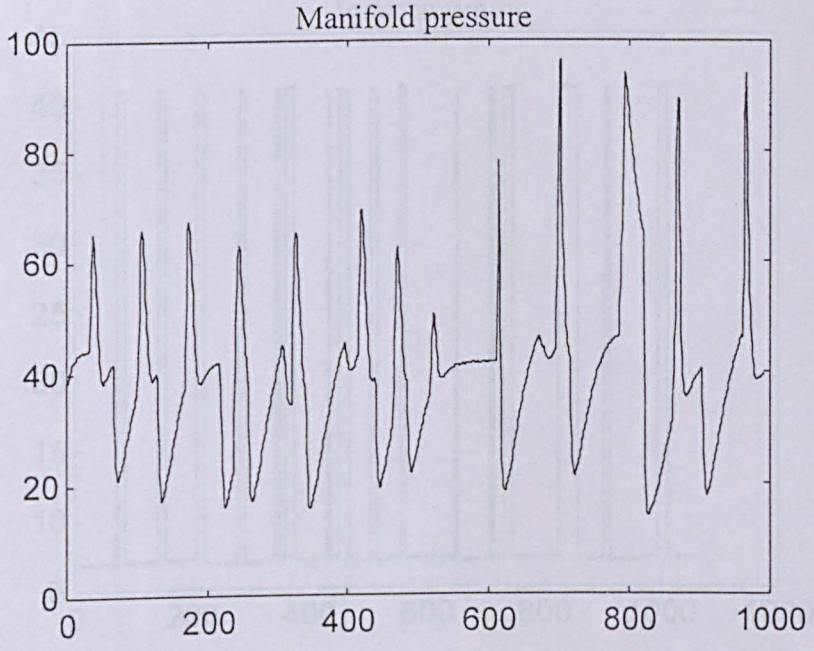


Figure 4.2b Manifold pressure (estimated model)

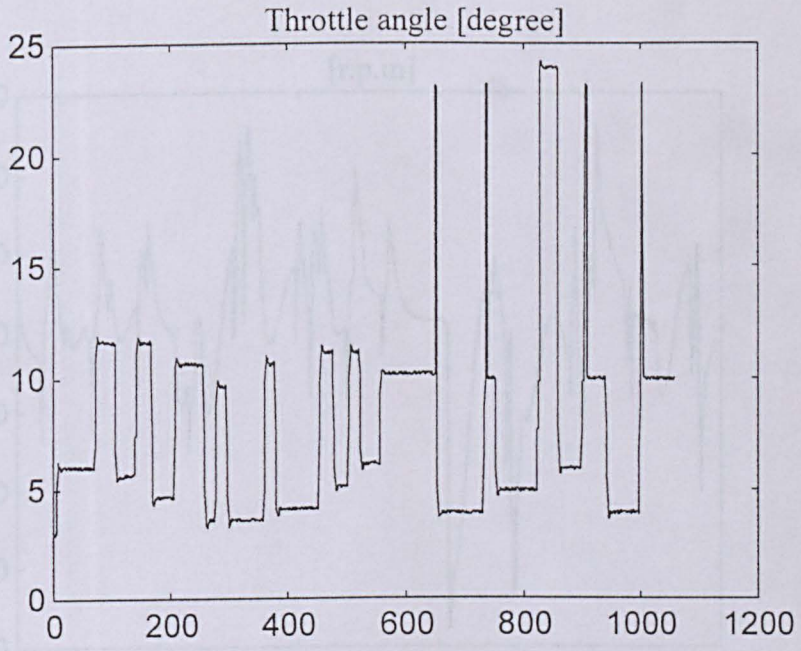


Figure 4.3 Throttle angle

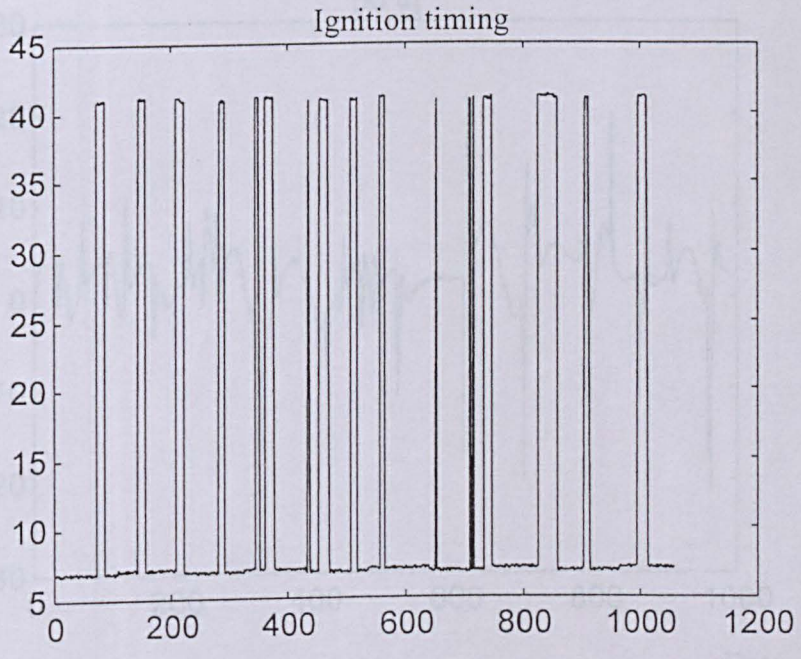


Figure 4.4 Ignition timing

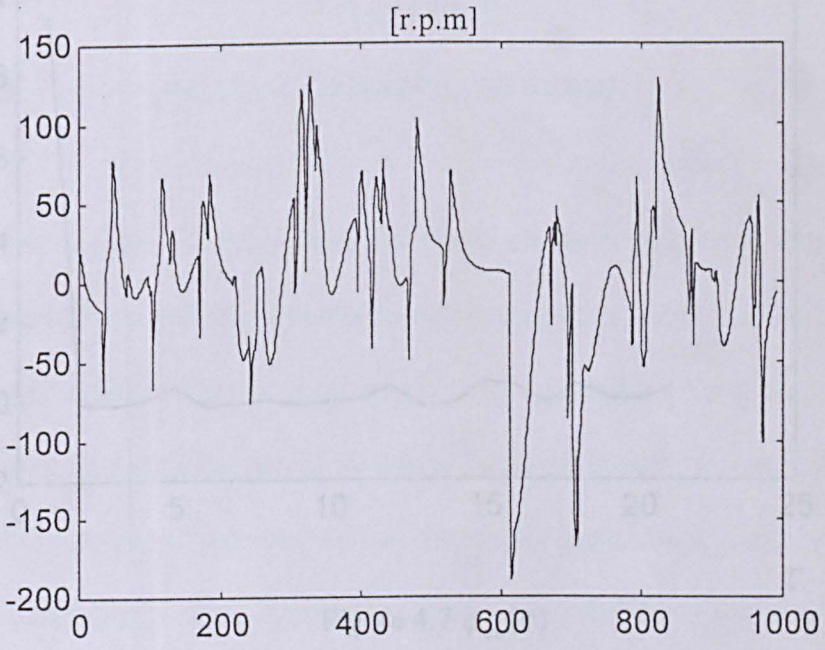


Figure 4.5 Speed error

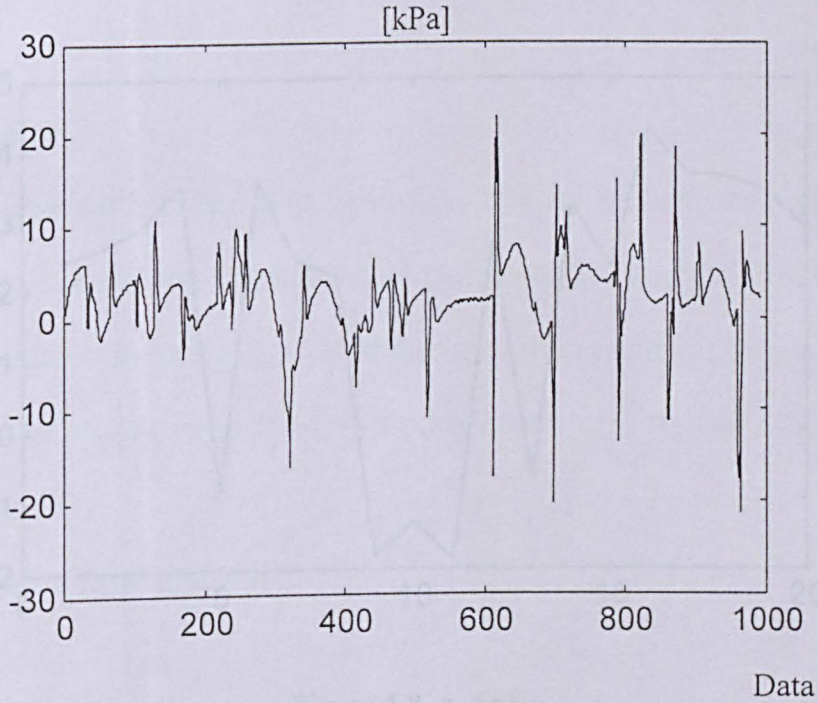


Figure 4.6 Manifold pressure error

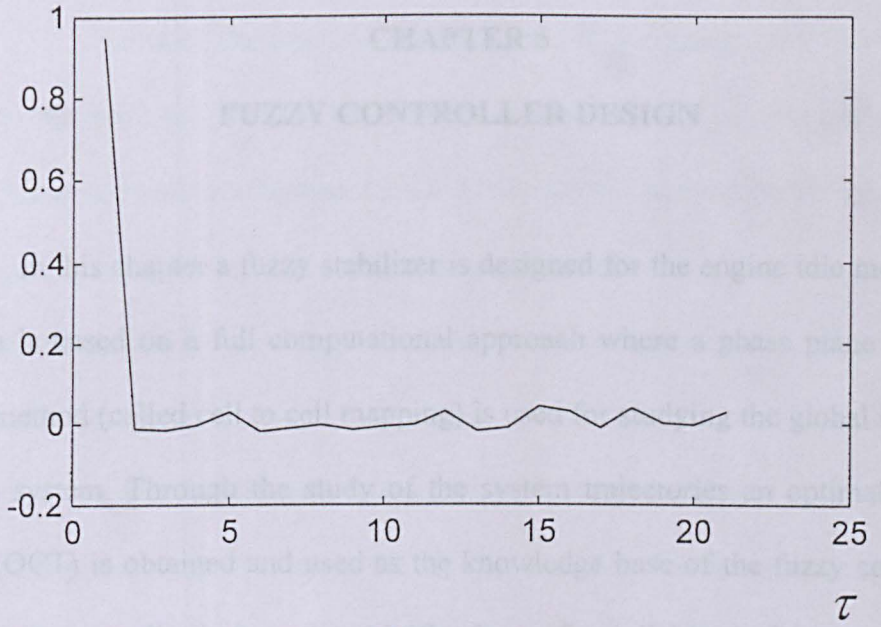


Figure 4.7 $\varphi_{\xi\eta}(\tau)$

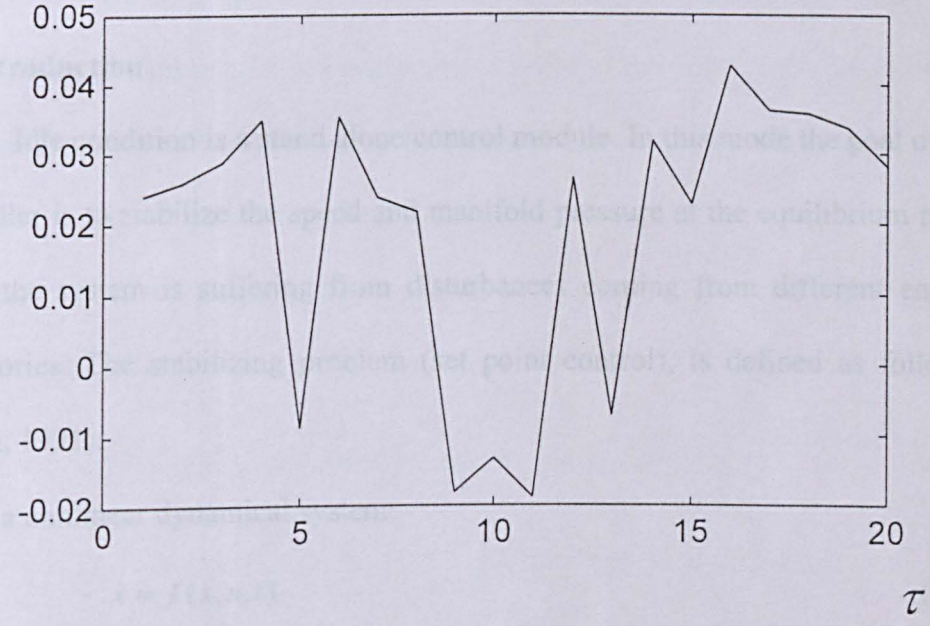


Figure 4.8 $\varphi_{\theta\eta}(\tau)$

CHAPTER 5

FUZZY CONTROLLER DESIGN

In this chapter a fuzzy stabilizer is designed for the engine idle mode. The design is based on a full computational approach where a phase plane analysis alike method (called cell to cell mapping) is used for studying the global behavior of the system. Through the study of the system trajectories an optimal control table (OCT) is obtained and used as the knowledge base of the fuzzy controller. The fuzzy controller is formulated in the form of a radial basis function network where the entries of the optimal control table are used as training data for estimating the controller design parameters.

5.1 Introduction

Idle condition is a stand alone control module. In this mode the goal of the controller is to stabilize the speed and manifold pressure at the equilibrium point while the system is suffering from disturbances coming from different engine accessories. The stabilizing problem (set point control), is defined as follows, (Slotin, 1991):

Given a nonlinear dynamical system:

$$\dot{x} = f(x,u,t) \tag{5.1}$$

where f is a nonlinear vector function, x is the state vector, u is the control vector and t is the time, find a control law such that starting from anywhere in a region Ω in the neighborhood of the equilibrium point; $x \to 0$ as $t \to \infty$.

Some of the common methods used in designing controllers for nonlinear systems are the phase plane method, Lyapunov stability analysis and describing function methods (Slotine,1991). Phase plane analysis relies on studying the global behavior of the autonomous point map $\dot{x} = f(x,u,t)$. Global behavior means the evolution of system states corresponding to various initial conditions.

Analyzing the global behavior of the system through numerical evaluation of the point map (Equation 5.1) is inefficient and time consuming. Instead Hsu (1980a) proposed the concept of "cell to cell mapping" (CCM) to study the global behavior of nonlinear dynamic systems. In this method the state space of the system is handled as a group of cells rather than a quantum of points. In the method the state space is discretized to a group of disjoint sets which are called cells, each center of a cell represents a value of the state vector at that point. The interval in which a value of a state variable x_i ($i=1,2,...,N$) is located is defined by:

$$(Z_i - 0.5)h_i \le x_i < (Z_i + 0.5)h_i$$

where Z_i is an integer and h_i is the interval size. A vector cell z is defined by the N-tuple Z_i where:

$z = \sum_{i=1}^{i=N} Z_i e_i$ where e_i is the unit vector in the Z_i - direction

$i=1,2,...,N$. A point x (x_i , i=1,2,...,N) belongs to a cell z (Z_i=1,2,...,N) if x_i belongs to Z_i for all i. Now the system state space can be seen as a collection of cells rather than a quantum of points where the number of the cells is integer.

The mapping, which represents the system can be written in the form, (Hsu,1985)

$$z(n) = C(z(n-1), u(n), \tau(n)) \qquad (5.2)$$

$z \in Z^m$, Z^m is the set of integers , $u(n)$ is the control vector at n time step.

C is the mapping rule, which maps a set of integers to a set of integers.

Equation (5.2) represents the nth step mapping from $z(n-1)$ to $z(n)$ under the action of $u(n)$ during the time interval $\tau(n)$. $z(n-1)$ is called the domain cell while $z(n)$ is called the image cell. The image cell is evaluated by integrating the corresponding point mapping over the time interval $\tau(n)$. As the cell size shrinks, the CCM can preserve the qualitative nature of the point map. A large cell size leads to increased cumulative errors while smaller cell size produces better accuracy but more memory is required for the computations. The CCM can be used to study the evolution of the state trajectories and detect equilibrium points ($\dot{x} = 0$ in point map and $z=C(z)$ in CCM (called a *P-1* motion)), periodic motions (limit cycles (called *P-k* motion)) and their domain of attraction (Hsu,1980b). An advantage is that it is applicable to systems of large dimensions more than two.

CCM was used to develop an optimal control algorithm for set point controllers (Hsu,1985). These controllers rely on an optimal control table (OCT) which contains the near to optimum control action for each cell in the cell space (direct digital controllers). As the system trajectory moves from one cell to another, the optimum control signal is looked up from the optimal control table and applied to the system. These controllers require a large memory size and their action is leading to the problem of chattering.

Smith (1991, 1992 and 1994) matched the OCT derived by the CCM to a fuzzy controller. The result is a reduced memory controller with smooth transitions without chattering. In this work the methodology of Smith (1991, 1992

and1994) with some alterations (to suit the multi input output nature of the problem in simulation conditions and controller parameters estimation) to match the problem being considered, is used to design a fuzzy stabilizer for the engine idle mode.

5.2 Obtaining the Optimal Control Table

In the idle mode control problem we have a two dimensional space of states x_1, x_2 ,where x_1 is the speed and x_2 is the manifold pressure. Determination of the equilibrium point is done by applying the basic idea of the unraveling algorithm (Hsu, 1980). Equilibrium points are characterized by their zero gradient of the state vector ($\dot{x} = 0$ in point map). In the cell space the equilibrium is defined by: $z=C(z)$ (called a *P-1* motion). An equilibrium point is detected by evaluating the model equation until the simulation time becomes very large and the state trajectories cannot penetrate the next cells, or the distance travelled in each direction is too small to exit the cell.

In this case the speed is required to converge to the value of 750 r.p.m with a pressure where the engine can run in a stable way without any fluctuation in the state (i.e. $z= (z)$ (a *P-1* motion)). The no load equilibrium point was found to be $N=750$ r.p.m and $P=30$ kPa (no load condition). In the load condition the equilibrium point is allowed to move in the direction of the state x_2 (manifold pressure) as near as possible to the value of 30 kPa, but still at the same point for x_1 (750 r.p.m). The control goal is to keep the engine at the equilibrium point when it is in its idle mode and suffering disturbances coming from different

engine accessories. The control signals are the throttle angle in the interval {5 25} and the ignition timing interval is {7 41}while the load is in the interval {0 60}N.m.

The state space of the system is discretized to intervals of variable size which are fine near the equilibrium point and coarse far from it (Smith (1994)). The load is also discretized into intervals.

All possible control actions and load are applied to all possible initial conditions of interest. The possible initial conditions of interest are in the interval {450 1150} for the speed (state x_1) and {11 105} for the pressure (state x_2). The possible control actions are {5 25} for the control action u_1 and {7.5 42} for the control action u_2 while the load is in the interval {0 60}.

Simulation of the plant is run for a number of times equal to the number $n_1 \times n_2 \times n_3 \times n_4 \times n_5$ where n_1 is the number of cells in the x_1 direction and n_2 is the number of cells in the x_2 direction, n_3 is equal to the number of control actions u_1 , n_4 is equal to the number of control actions u_2 , n_5 is the number of load intervals. After running the simulation a search algorithm is applied to extract the optimal control table (OCT) from the simulation results.

5.2.1 Obtaining the state trajectories

The two dimensional space is quantized to the following cells:

[450 550 600 650 700 750 800 850 900 1000 1100 1150], for the first state x_1 (12 intervals).

[11.4286 22.8571 34.2857 45.7143 57.1429 68.5714 80 91.4286 105], for the second state x_2 (9 intervals).

cells are not considered. A sorting algorithm must be applied to sort the results to figure out the optimum cell transitions i.e. the optimum control table OCT. Details of the sorting algorithm can be found in Hsu (1985).

5.2.2 Search procedure

The resulting table must be sorted to figure out the optimum path of each cell to the target cell (equilibrium point). We apply a rough copy of the algorithm in Hsu (1985) in the following steps

1- Starting from the target cell find the next cells to it which have a direct transition to the target cell, then put them in a set A.
2- Each cell in the set A has different paths to the target cell from these paths the one with the minimum cost function is taken (Figure 5.1).
3- For each cell in the set A find a set B_i which contains cells having a path to a cell in A, now each cell in A is considered as a target cell.
4- For each set B_i apply step 2.
5- Each cell in a set B_i is now considered as a target cell and the above procedures are applied to it.
6- Keep back propagating in this manner until the whole data set is exhausted.

After the search steps are applied we have a set of cells transitions and the corresponding optimal control action which can drive them to the target cell, which is called the optimal control table. The optimal control table itself can be used as a direct digital controller with its known disadvantages. Instead this OCT

can be used as the source of knowledge of the fuzzy controller. The entries of the table will be used as a training data to estimate the parameters of the fuzzy controller in the next section.

5.3 Construction of the Fuzzy Controller

Typical fuzzy controllers consist mainly of a rule base, a fuzzy inference engine, input interface (fuzzifier) and output interface (defuzzifier).

The rule base is constructed from a collection of IF-then rules and the inference engine operates on the rule base to determine a mapping from fuzzy sets in the input universe of discourse $\chi \subset R^n$ to a fuzzy set in the output universe of discourse $V \subset R^m$.

A fuzzy rule base has the form:

R^1: *IF* x_1 *is* f_1^1 *and* x_2 *is* f_2^1 . . . *and* x_n *is* f_n^1 *THEN* u_1 *is* G_1^1 *and* u_2 *is* G_2^1 . . . and u_m *is* G_m^1

.

.

R^l: *IF* x_1 *is* f_1^l *and* x_2 *is* f_2^l . . . *and* x_n *is* f_n^l *THEN* u_1 *is* G_1^l *and* u_2 *is* G_2^l . . . and u_m *is* G_m^l

.

.

R^M: *IF* x_1 *is* f_1^M *and* x_2 *is* f_2^M . . . *and* x_n *is* f_n^M *THEN* u_1 *is* G_1^M *and* u_2 *is* G_2^M . . . and u_m *is* G_m^M

where $l = 1, 2, \ldots, M$ and M is the number of rules, f_i^l, G_j^l are fuzzy sets and $x_1, x_2, \ldots, x_n$ are the states of the plant and $u_1\ u_2, \ldots u_m$ are the control actions

The fuzzy inference engine operates on the fuzzy IF-THEN rule to perform a mapping:

$$f_1' \times f_2' \times \ldots \times f_n' \rightarrow G_1' \times G_2' \times \ldots \times G_m' \tag{5.3}$$

where, G_1', G_2', . . . , G_m' are fuzzy sets in $\chi \times U$ and $\times$ is the composition operator. Rewriting the previous rule in the implication form:

$$A \rightarrow B \tag{5.4}$$

There are many methods for interpreting the previous implication. The most common of them are the minimum (*min*) operation rule of fuzzy implication:

$$\mu_{A\rightarrow B}(x,u) = \min\{\mu_A(x).\mu_B(u)\} \tag{6.5}$$

The product operation rule of fuzzy implication:

$$\mu_{A\rightarrow B}(x,u) = \mu_A(x),\mu_B(u) \tag{5.6}$$

The fuzzifier performs a mapping from a crisp point $x = (x_1, x_2, \ldots x_n)$ into a fuzzy set A. The fuzzifier may be of the singletone type:

$$\mu_A = \begin{cases} 1 & \text{if } x' = x \\ 0 & \text{if } x' \neq x \end{cases} \tag{5.7}$$

or a nonsingletone type (Lee(1990)):

$$\mu_A = \begin{cases} 1 & \text{if } x' = x \\ \exp[-(x'-x)^T(x'-x)/\sigma^2] \end{cases} \tag{5.8}$$

The defuzzifier performs a mapping from fuzzy sets in U to crisp point in V. The most common defuzzifier is the center average defuzzifier:

$$u = \frac{\sum_{l=1}^{M} u^l \mu_B^l(u^l)}{\sum_{l=1}^{M} \mu_B^l(u^l)} \tag{5.9}$$

The whole mapping of the fuzzy logic system (Wang, 1994) with singleton fuzzifier, product inference rule and center average defuzzfier can be represented by:

$$u = \frac{\sum_{l=1}^{M} u^l \prod_1^n \mu_B^l(u^l)}{\sum_{l=1}^{M} \prod_1^n \mu_B^l(u^l)} \tag{5.10}$$

If the gaussian function is taken as the membership function then:

$$u = \frac{\sum_{1}^{M} u^l \prod_{1}^{n} \exp[-(x'-x_c)^T(x'-x_c)/\sigma^2]}{\sum_{1}^{M} \prod_{1}^{n} \exp[-(x'-x_c)^T(x'-x_c)/\sigma^2]} \tag{5.11}$$

The fuzzy logic system (Equation 5.11) has three design parameters which are: u^l, x_c and σ_i^l. These parameters are design parameters of the fuzzy logic controllers, they can be trained to find the optimum values for them.

Rewriting Equation (5.10) in the form:

$$u = \sum_{j=1}^{M} p_j(x)\theta_j \tag{5.12}$$

Where

$$P_j(x) = \frac{\prod_1^n \mu_B^l(u^l)}{\sum_{l=1}^{M} \prod_1^n \mu_B^l(u^l)}$$

$j=1,2,\ldots, M.$

p_j is called fuzzy basis function (FBF) (Wang, 1994).

The expansion (5.12) (called the fuzzy logic expansion) with the definition of p_j and the gaussian function (which has a radial structure) is equivalent to the expansion (3.6) of Chapter 3, which can be clearly represented as radial basis expansion.

The fuzzy logic system can be represented as a radial basis function alike network. This network can be trained using the OLS algorithm used in Chapter 3. In this work OLS algorithm was used to train the system. The input and output values are the entries of the optimum control table derived before. The inputs are the states of the plant x_1 , x_2 and the load. The outputs are the control actions u_1 , u_2 . The same treatment of Equation (3.6) is used here to solve the problem of finding the set of centers x_c and the set of parameters u^l .

The OLS algorithm code has to be modified to match the new problem, Equation (5.12) is rewritten in the matrix form:

$$u = \mathrm{P} \tag{5.13}$$

$u = [u(1), u(2), \ldots, u(N)]$, $\mathrm{P} = [p_1, p_2, \ldots, p_M]$, $p_i = [p_i(1), p_i(2), \ldots, p_i(N)]^T$

The OLS algorithm transforms the set of p_i into a set of orthogonal basis functions, and uses only the significant vectors to form the final FBF expansion. The algorithm steps will be as follows:

Step 1

For $1 \le i \le N$, compute

$$w_1^{(i)} = p_i \ , \ g_1^i = (w_1^i)^T \ d^o / ((w_1^i)^T w_1^i)$$

$$[err]_1^i = (g_1^i)^2 (w_1^i)^T w_1^i / (d^{oT} d^o)$$

Find

$$[err]_1^{(i_1)} = \max([err]_1^i \ , \ 1 \le i \le N)$$

select $\quad w_1^{i1} = w_i^{(i_1)} = p_{i_1} \ , \quad g_1 = g_1^{i1}$

kth Step

$$\alpha_{jk}^i = w_j^T p_i / (w_j^T w_j) \ , \ 1 \le j < k$$

$$w_k^i = p_i - \sum_{j=1}^{k-1} \alpha_{jk}^i w_j \ , \qquad g_k^i = (w_k^i)^T d^o / ((w_k^i)^T w_k^i)$$

$$[err]_k^{(i_k)} = \max([err]_k^i \ , \ 1 \le i \le N, \ i \ne i_1, \ldots, i \ne i_{k-1})$$

select $w_k = w_k^i$, $g_k = g_k^i$

Solve the triangular system:

$$A^{M_s}\theta^{M_s} = g^{M_s}$$

$$A^{M_s} = \begin{bmatrix} 1 & \alpha_{12}^{i2} & \alpha_{13}^{i3} & \dots & \alpha_{1M_s}^{iM_s} \\ 0 & 1 & \alpha_{23}^{i3} & \dots & \alpha_{2M_s}^{iM_s} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & \alpha_{M_{s-1},M_s}^{iM_s} \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$g^{M_s} = [g_1, \dots, g_{Ms}]^T \ , \quad \theta^{M_s} = [\theta_1^{Ms}, \dots, \theta_{M_s}^{M_s}]^T$$

The final FBF expansion is

$$u = \sum_{j=1}^{M_s} p_{ij}(x)\theta_j^{M_s}$$

The same code used for modeling in Chapter 3 is also used here with some little modification to suit the problem discussed here.

For the load condition the set of centers is given by the matrix C^T,

$$C^T = \left[\frac{C_1^T}{C_2^T}\right]$$

while the set of parameters is given by P :

$$P = \left[\frac{P_1}{P_2}\right] \tag{5.13}$$

For the no load condition:

$$p = \left[\frac{p_1}{p_2}\right]$$

$$c^T = \left[\frac{c_1^T}{c_2^T}\right] \tag{5.14}$$

C_1^T, C_2^T, P_1, P_2, c_1^T, $c'_{2,}$, p_1, and p_2 are given on pages 90-95 .

5.4 Simulation, Results and Discussion

A block diagram of the fuzzy controller and the plant is shown in Figure 5.2. The controller has two control modes, a mode under no load condition and a mode under load condition In the load condition the controller has three input signals, which are; the speed, manifold pressure, and the load and two output signals, which are the throttle angle and the ignition timing. In the no load condition the inputs to the controller are the two states(P and N) and the outputs are the throttle and ignition timing signals.

The idle mode controller designed here starts control at speed 1150 [r.p.m] when the engine is decelerating to the idle condition. In Figure 5.3 a,b the response curves for the speed and pressure under no load condition are shown with the control signals in Figure 5.3c,d and the state trajectories in Figure 5.3e. Control performance in no load condition for different initial conditions is shown in Figures 5.3 to 5.4 where we can see that the states of the system are converging to the equilibrium point (750 r.p.m,30 kPa).

The number of rules in the no load mode is 34 while in the load condition is 76. The number of rules depends mainly on the number of cells in the domain of interest, (in other words depends on the speed at which the idle mode control is enabled). If we reduce the domain of interest of the cells in the no load or load conditions the number of rules will be reduced. For example the number of rules

required to control the idle mode under load condition at 750 r.p.m is 14 rules (if the idle mode is enabled at 850 r.p.m). We prefer to enlarge the domain of cell's space and consequently the number of rules to compensate for variations of the plant parameters over its time life and unexpected operating conditions (speed or pressure overshoots). A second reason is the nature of fuzzy controller, as the fuzzy controller is nothing more than a predefined associative memory (FAM) which cannot extrapolate unless it is designed to be a self organizing or adaptive. In other words, for our case (where operating conditions and plant parameters can abruptly change) the control strategy will be more robust and reliable with increased number of rules.

Convergence to the equilibrium and dynamic response characteristics (especially the settling time) of the plant were severely affected by the spread factor of the gaussian membership function. The spread factor determines the degree of overlap between the labels of the rule base. It can severely affect the performance of the designed fuzzy controller in the load mode. That is mainly because of two reasons. The first reason is that the designed controller is a multi input multi output fuzzy controller (three input variables and two output control actions). The second reason is the rough quantization of the third input variable (torque) in trying to reduce the number of rules. The spread factor has been tuned when the state trajectories are approaching the equilibrium point under different loading conditions. The method of choosing this parameter proposed by Wang (1994) was not practical in our case. However the same spread factor was initially used for the whole rule base. Later on, it was tuned on-line under the load condition so the states can converge to the equilibrium point.

Simulation under load condition is shown in Figures 5.5, where in all cases the plant states still converge to the required equilibrium points of the speed with higher manifold pressure. In all situation the states of the system converge to the equilibrium point or to a very small region Ω around it.

Quantitative analysis of nonlinear control systems behavior is not so obvious like the one of linear control systems. For the linear type it is possible to systematically specify some specifications like settling time, overshoot and rise time. For a nonlinear control system the situation is different as the system response for an input signal is different from the response to another input signal and a frequency response domain analysis is not possible. Instead some qualitative specifications such as stability, accuracy and speed of response, robustness and cost are used.

For the fuzzy control system designed here it can be seen that system is globally stable because over all its controllable space, the system states converge to the equilibrium point and stay in Ω. Accuracy of the outputs responses can be seen in all Figure 5.3a,b,e and Figure 5.4a,b,e where the speed response converges to 748 to 752 of a nominal value 750, the manifold pressure converges to 29.6 kPa of a nominal value 30 kPa. The speed of response can also seen from the same figure where in all cases the states settle in less than 0.4 second.

As a proof of the robustness of the control system the parameters of the plant are changed with small percentages. It can be seen from Figure 5.6a,b,e. that the states of the plant still converge roughly to the neighborhood of the equilibrium point. From the above discussion it can be concluded that the controller developed here performs efficiently as can be seen from the simulation

results. It can compensate for plant uncertainties and drive the states of the system accurately to their equilibrium point.

Controller parameters are given in the following matrices:

$$C_1^r = \begin{bmatrix} 0.9190 & 0.5087 & 0.6444 \\ 0.7468 & 0.7530 & 0.6440 \\ 0.8356 & 0.6261 & 1.0086 \\ 0.9186 & 0.2552 & 0.3205 \\ 1.0022 & 0.3776 & 1.0082 \\ 0.9148 & 0.3810 & 0.6498 \\ 0.7473 & 0.6344 & 0.3220 \\ 0.8345 & 0.3827 & 0.3277 \\ 0.8344 & 0.6341 & 0.6459 \\ 0.7441 & 0.8782 & 1.0067 \\ 0.6545 & 0.5024 & 1.0069 \\ 0.6589 & 0.8833 & 0.6500 \\ 0.8327 & 0.3771 & 1.0002 \\ 0.6588 & 0.8763 & 0.3238 \\ 0.6597 & 0.2572 & 1.0047 \\ 1.0061 & 0.1254 & 0.3233 \\ 0.6547 & 0.2548 & 0.6497 \\ 0.9205 & 0.5027 & 1.0011 \\ 0.8309 & 0.7585 & 1.0052 \\ 0.9131 & 0.2577 & 0.6486 \\ 0.9196 & 0.2555 & 1.0004 \\ 0.7481 & 0.7531 & 0.3209 \\ 0.7485 & 0.8761 & 0.6499 \\ 0.8358 & 0.5021 & 0.3206 \\ 0.6587 & 0.8791 & 1.0057 \\ 0.6599 & 0.5029 & 0.6487 \\ 0.6546 & 0.2513 & 0.3270 \\ 0.6544 & 0.6318 & 1.0082 \\ 0.6527 & 0.2556 & 0.1635 \\ 0.6583 & 1.0008 & 1.0016 \\ 0.6525 & 0.3839 & 1.0093 \\ 0.7394 & 0.3807 & 0.6423 \\ 0.6574 & 0.5083 & 0.1614 \\ 0.9143 & 0.1345 & 0.3263 \\ 0.6591 & 0.5053 & 0.3249 \\ 0.8273 & 0.7566 & 0.6482 \\ 0.8298 & 0.5039 & 0.6475 \\ 1.0055 & 0.5033 & 0.6447 \\ 0.9174 & 0.6323 & 0.6403 \\ 0.6572 & 0.6258 & 0.6431 \end{bmatrix}$$

$$C_2^T = \begin{bmatrix} 1.0061 & 0.2550 & 0.3280 \\ 0.8297 & 0.2550 & 0.3215 \\ 0.9226 & 0.3786 & 0.3204 \\ 0.8273 & 0.5096 & 1.0056 \\ 0.6598 & 0.3765 & 0.3271 \\ 0.6578 & 0.7532 & 1.0054 \\ 1.0021 & 0.2542 & 0.6423 \\ 0.9146 & 0.6314 & 1.0010 \\ 1.0023 & 0.5080 & 1.0016 \\ 1.0037 & 0.2551 & 1.0095 \\ 0.8266 & 0.2548 & 1.0052 \\ 0.6599 & 0.6320 & 0.1692 \\ 0.6566 & 0.3835 & 0.1607 \\ 0.8269 & 0.2589 & 0.6452 \\ 0.6558 & 0.3800 & 0.6485 \\ 0.6533 & 1.0084 & 0.3220 \\ 0.6590 & 1.0053 & 0.6464 \\ 0.8330 & 0.3823 & 0.6403 \\ 1.0011 & 0.3795 & 0.6444 \\ 0.7489 & 0.2542 & 0.6433 \\ 0.6567 & 0.6263 & 0.3226 \\ 0.7434 & 0.5094 & 0.3222 \\ 0.6597 & 0.7527 & 0.6415 \\ 0.7393 & 0.5061 & 1.0070 \\ 0.7417 & 0.2523 & 0.3252 \\ 0.7399 & 0.5046 & 0.6454 \\ 0.9147 & 0.3753 & 1.0034 \\ 0.7483 & 0.3770 & 1.0074 \\ 0.7453 & 0.2600 & 1.0067 \\ 0.6548 & 0.7557 & 0.3271 \\ 0.7451 & 0.6254 & 1.0002 \\ 0.7441 & 0.6287 & 0.6443 \\ 0.7416 & 0.3790 & 0.3293 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 0.6306 & 0.0346 \\ 1.6755 & 0.1194 \\ 1.2814 & 0.2421 \\ 0.4750 & 0.0471 \\ 1.4035 & 0.9942 \\ 0.9360 & 0.7310 \\ 1.5810 & 0.1899 \\ 1.1660 & 0.2705 \\ 1.0888 & 0.2217 \\ 1.4989 & -0.1604 \\ 0.3761 & 1.3127 \\ -0.1551 & 1.6961 \\ 0.3937 & 0.0592 \\ 1.2637 & 0.1670 \\ 1.3963 & -0.2967 \\ 1.3060 & 0.1946 \\ -0.2427 & 0.9847 \\ 0.4260 & 0.6745 \\ 0.4893 & 0.6471 \\ 0.3672 & 0.9873 \\ 0.3717 & 0.5247 \\ 0.2442 & 0.5966 \\ 0.4301 & 0.7442 \\ -0.0803 & 0.2010 \\ 0.2838 & 0.2910 \\ 0.7678 & 0.5868 \\ 0.4401 & 0.4926 \\ 0.7571 & 0.3795 \\ 1.4629 & 0.4306 \\ 0.9773 & 0.2854 \\ 1.0225 & 0.2272 \\ 0.3131 & 0.5227 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} 0.9353 & 0.5494 \\ 1.1859 & 0.5144 \\ 0.7482 & 0.2399 \\ 1.3637 & 0.3763 \\ 0.4287 & 0.5083 \\ 0.4221 & 0.4536 \\ 0.5733 & 0.3102 \\ 0.8200 & 0.1341 \\ 0.7458 & 0.2129 \\ 0.7341 & 0.5412 \\ 0.7115 & 0.3522 \\ 0.2797 & 0.1560 \\ 0.2691 & 0.1711 \\ 0.0991 & 0.2766 \\ 0.3409 & 0.5217 \\ 0.2400 & 0.0683 \\ 0.2518 & -0.0162 \\ 1.0943 & 0.0749 \\ 0.6870 & 0.0269 \\ 0.3516 & -0.1542 \\ -0.2019 & 0.1429 \\ -0.3117 & 0.0397 \\ -0.2273 & 0.0572 \\ -0.2174 & -0.1443 \\ -0.1841 & -0.1238 \\ -0.2552 & -0.1060 \\ -0.2126 & -0.1815 \\ -0.2481 & 0.0706 \\ 0.2055 & 0.0218 \\ -0.0853 & 0.0222 \\ -0.0210 & -0.1377 \\ -0.0041 & 0.1338 \\ -0.1293 & -0.0005 \end{bmatrix}$$

$$c_1 = \begin{bmatrix} 0.9213 & 0.1746 \\ 0.9143 & 0.3354 \\ 0.5694 & 1.0039 \\ 0.4829 & 0.5050 \\ 0.3916 & 0.6717 \\ 0.3970 & 0.3397 \\ 1.0076 & 0.6751 \\ 0.4800 & 0.8382 \\ 0.7491 & 0.1703 \\ 0.5736 & 0.6677 \\ 0.6557 & 0.3362 \\ 0.4798 & 1.0085 \\ 0.6573 & 0.5018 \\ 1.0077 & 0.8386 \\ 1.0094 & 0.3413 \\ 0.8324 & 0.3347 \\ 0.7422 & 0.6756 \\ 0.9132 & 0.5021 \\ 1.0046 & 0.1757 \\ 0.8348 & 0.1675 \\ 0.3966 & 0.8338 \\ 0.5737 & 0.3427 \\ 0.5679 & 0.8392 \\ 0.4811 & 0.6677 \\ 0.8333 & 0.6662 \\ 0.7415 & 0.5071 \\ 0.3993 & 0.5096 \\ 0.3963 & 1.0098 \\ 0.7480 & 0.3341 \\ 0.4836 & 0.3377 \\ 0.9199 & 0.6733 \\ 0.5693 & 0.5065 \\ 0.8334 & 0.5027 \\ 1.0005 & 0.5055 \\ 0.6581 & 0.6694 \end{bmatrix}$$

$$p_1 = \begin{bmatrix} 0.9406 & 0.1774 \\ 0.7701 & 0.1785 \\ 0.3386 & 1.0408 \\ 0.7444 & 1.7152 \\ 0.6780 & 0.9612 \\ 0.2543 & 1.3957 \\ 1.4418 & 0.1823 \\ 0.3090 & 0.8090 \\ 0.6173 & 0.2700 \\ 0.4391 & 0.6144 \\ 0.5245 & 0.4063 \\ 0.3880 & 1.0045 \\ 0.3686 & 0.6714 \\ 0.4801 & 0.1842 \\ 1.0888 & 0.1848 \\ 1.6705 & 0.1941 \\ 0.3554 & 0.2490 \\ 0.2997 & 0.2521 \\ 1.0199 & 0.2256 \\ 1.1617 & 0.1541 \\ 0.2133 & 0.7266 \\ 0.2428 & 0.4682 \\ 0.3688 & 0.1433 \\ 0.0363 & 0.4776 \\ 0.3354 & 0.1593 \\ 0.2834 & 0.0652 \\ 0.0518 & -0.3337 \\ 0.2663 & -0.0874 \\ -0.4087 & 0.1492 \\ -0.2370 & -0.0127 \\ -0.1140 & 0.1913 \\ 0.1200 & -0.2152 \\ 0.0034 & 0.2847 \\ 0.0467 & 0.1578 \\ 0.1367 & 0.1309 \end{bmatrix}$$

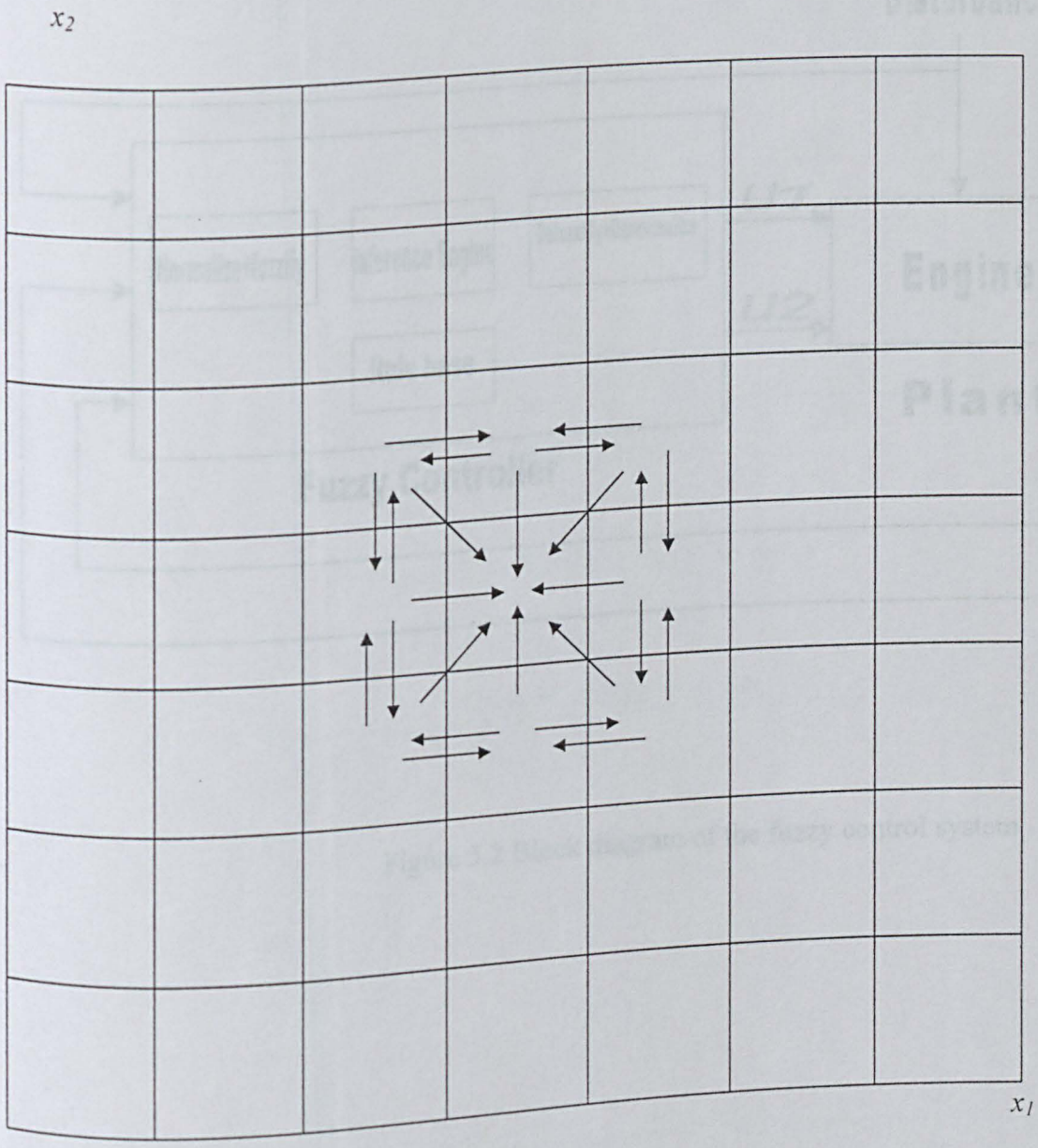


Figure 5.1 First step of the search

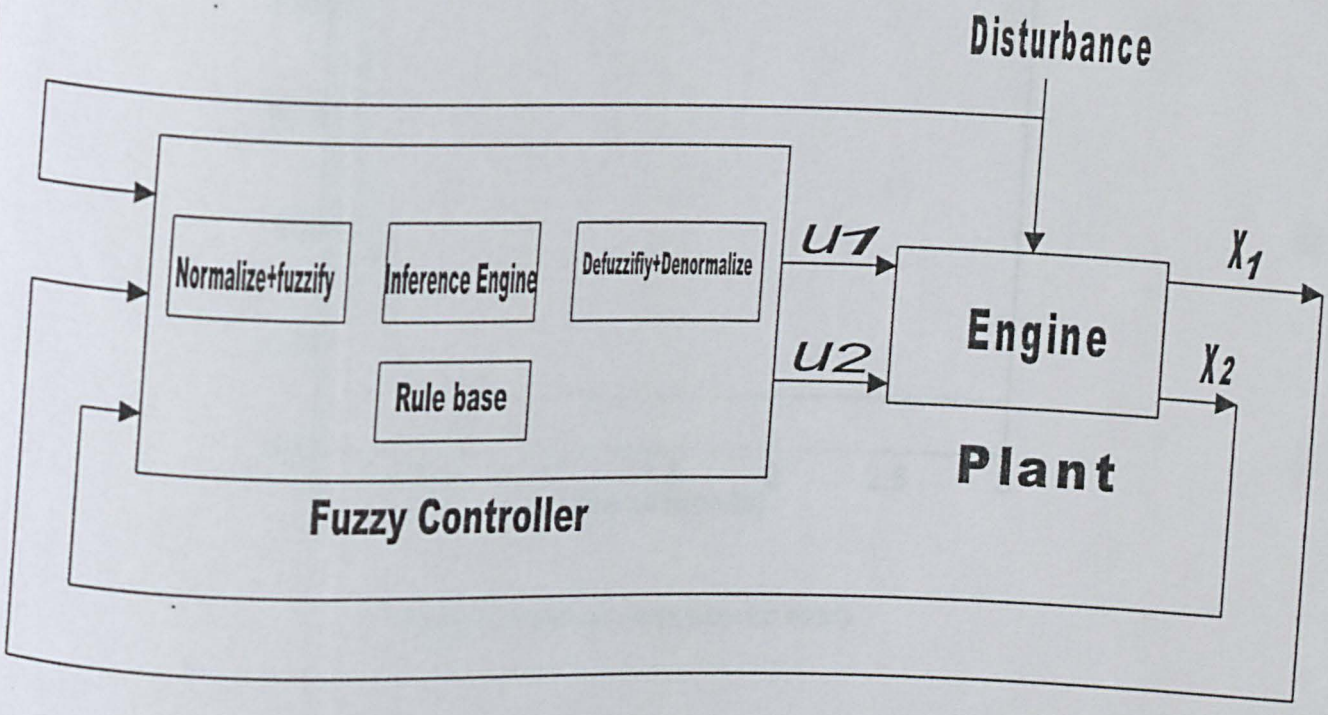


Figure 5.2 Block diagram of the fuzzy control system

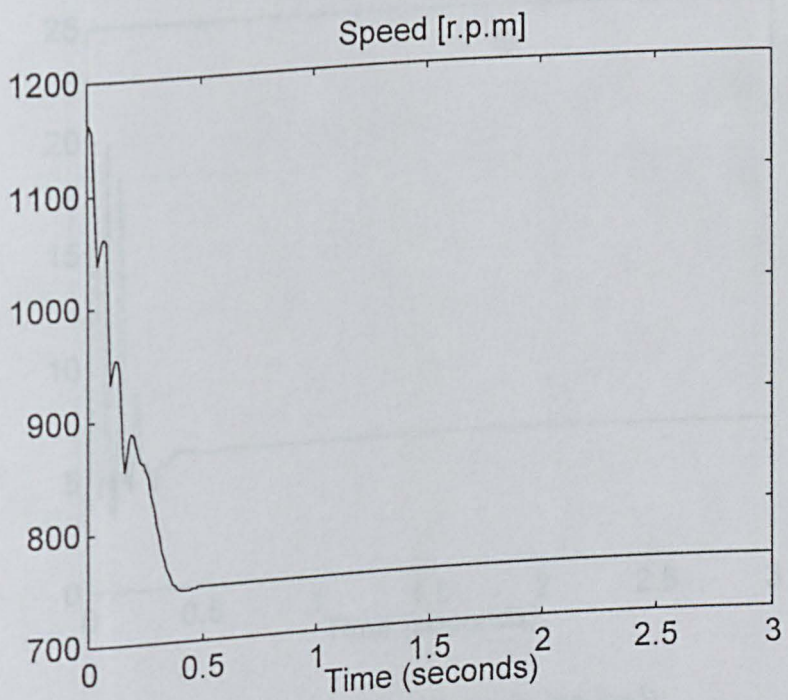


Figure 5.3a Speed response (no load)

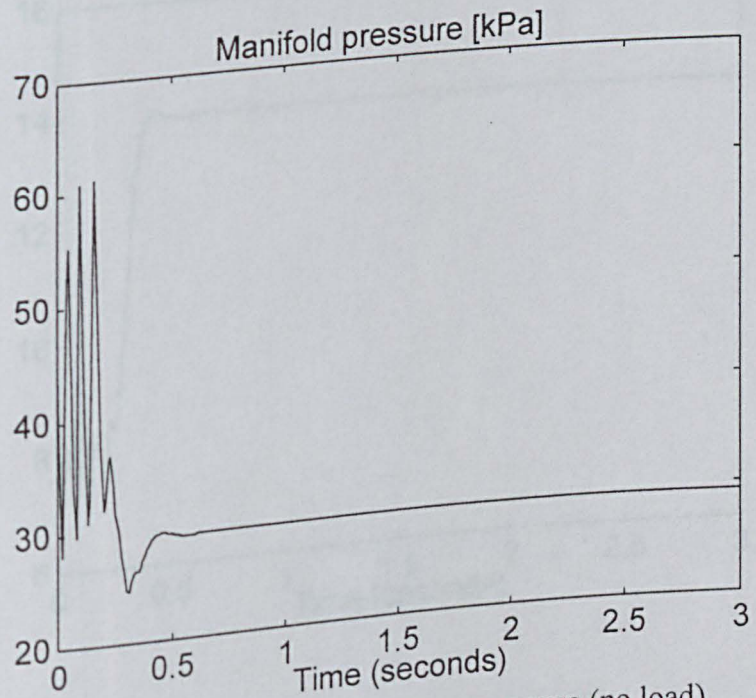


Figure 5.3b Manifold pressure response (no load)

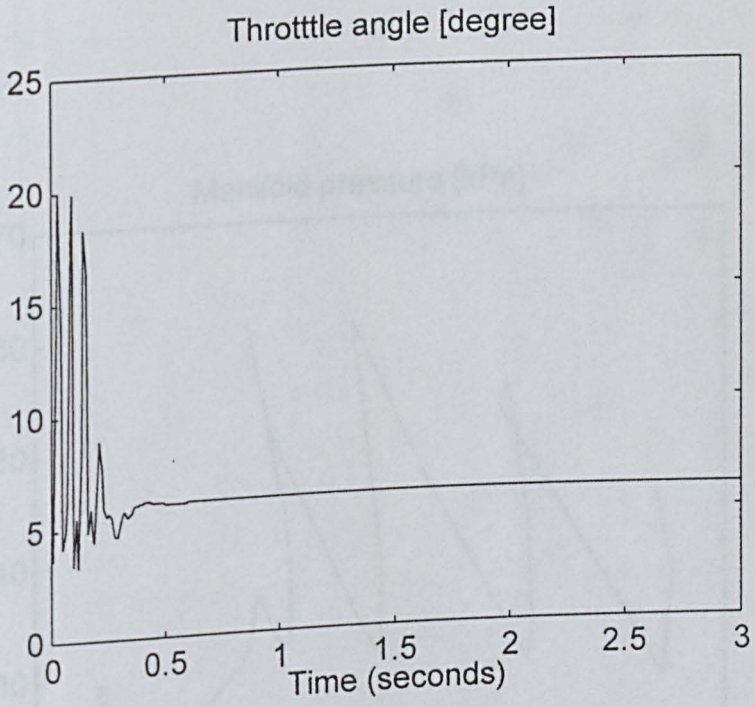


Figure 5.3c Throttle angle (no load)

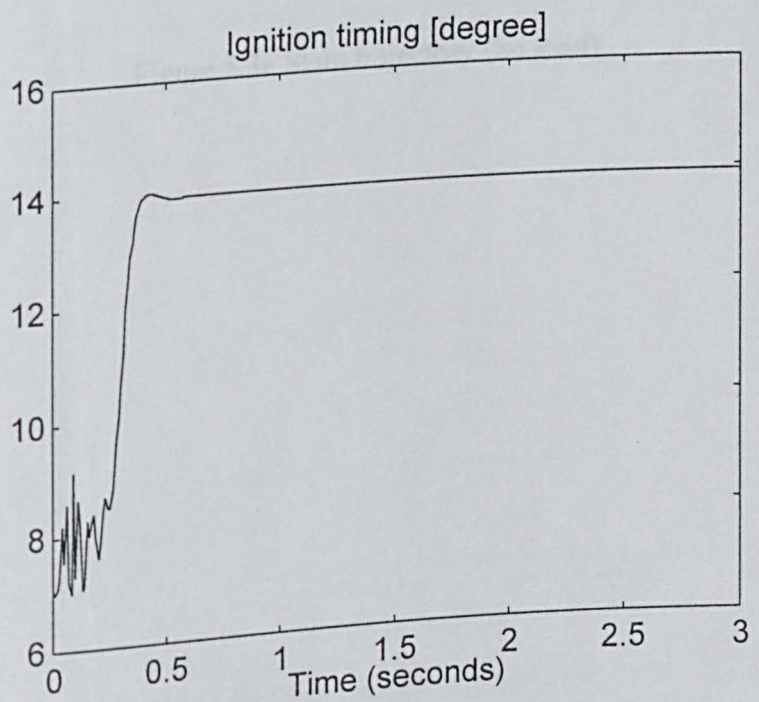


Figure 5.3d Ignition timing (no load)

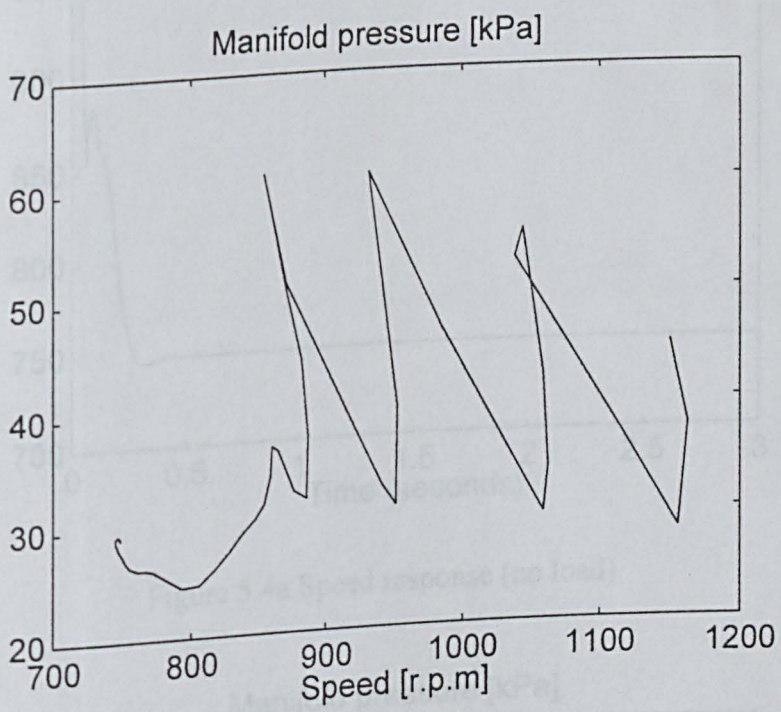


Figure 5.3e State trajectory (no load)

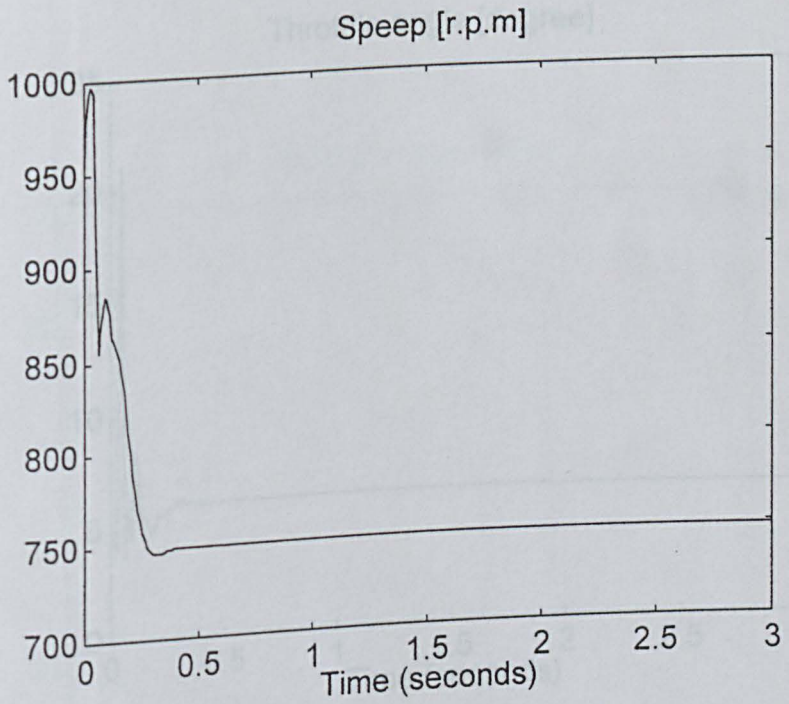


Figure 5.4a Speed response (no load)

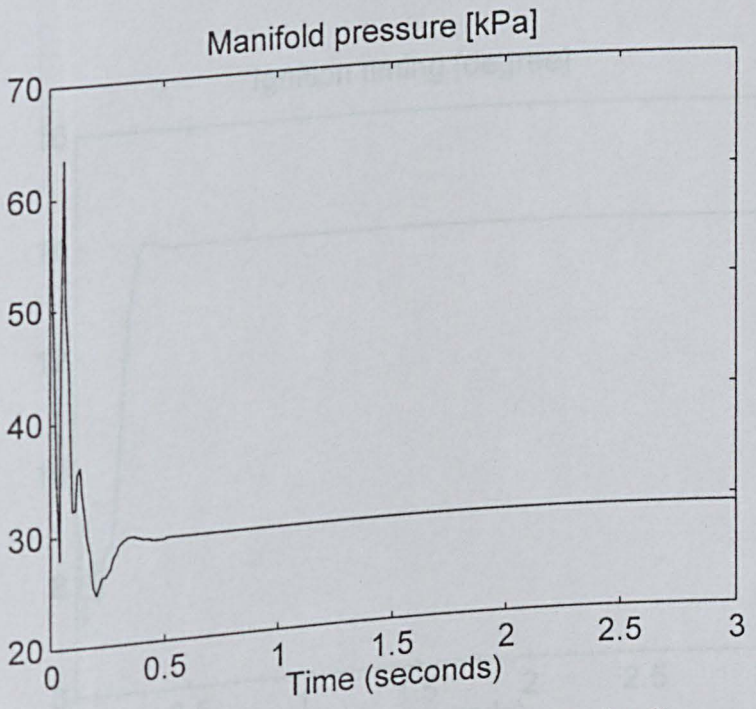


Figure 5.4b Manifold pressure response(no load)

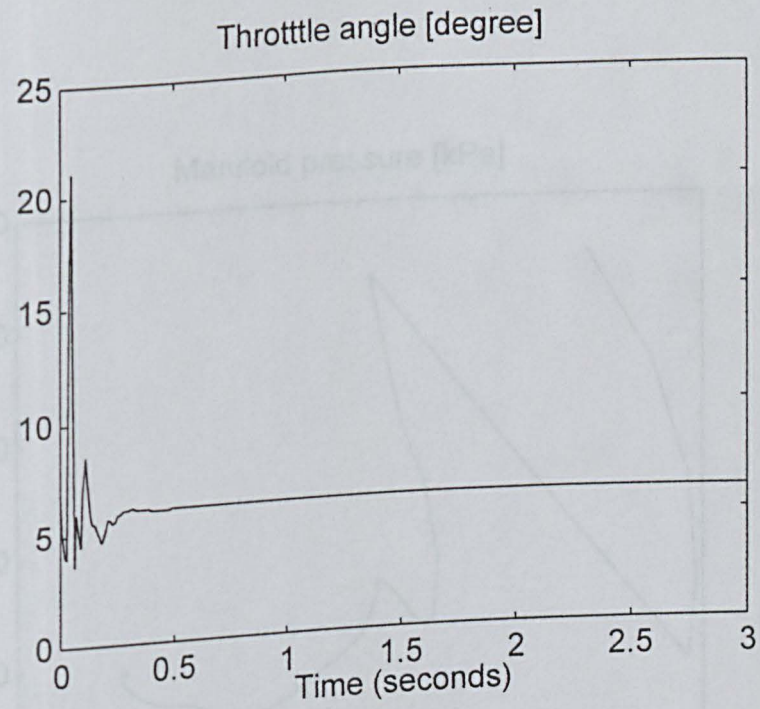


Figure 5.4c Throttle angle (no load)

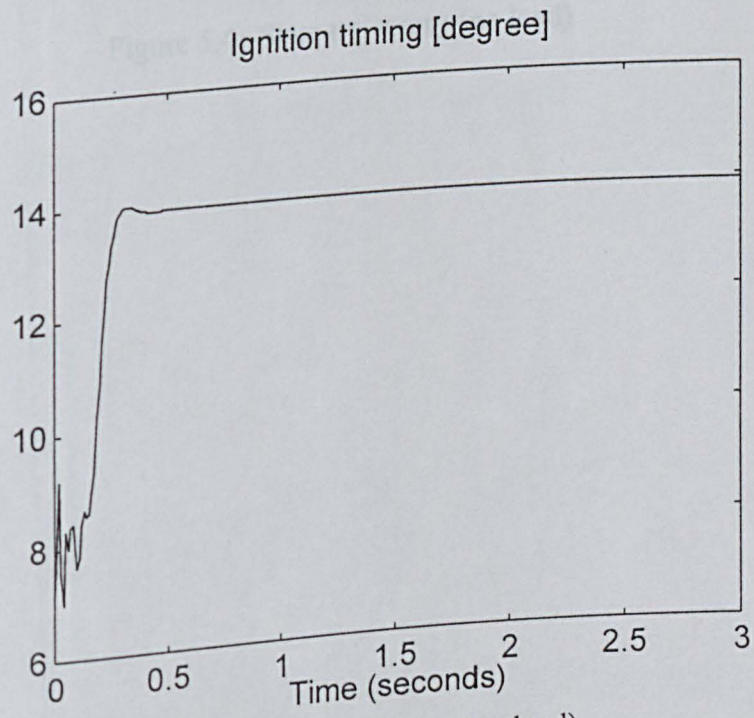


Figure 5.4d Ignition timing (no load)

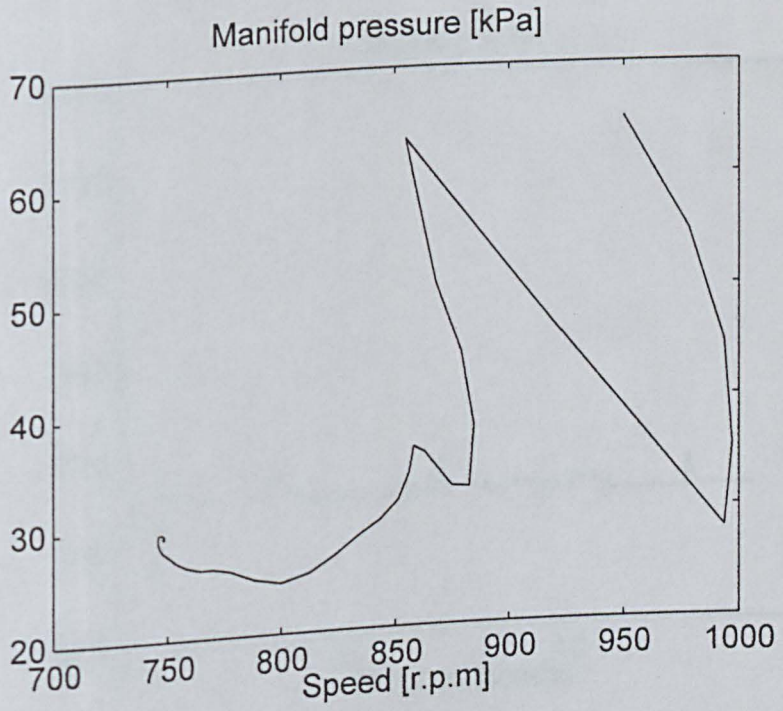


Figure 5.4e State trajectory (no load)

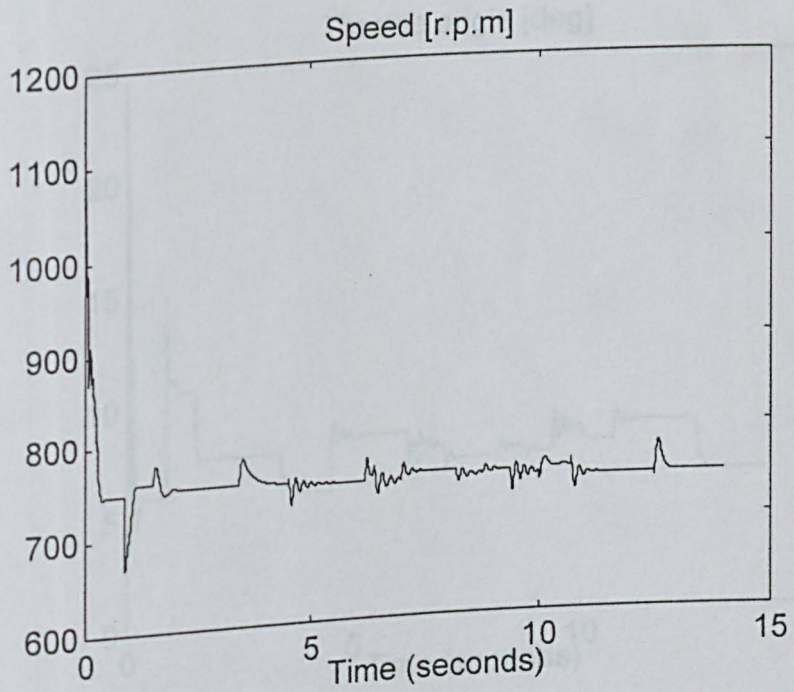


Figure 5.5a Speed response (under load)

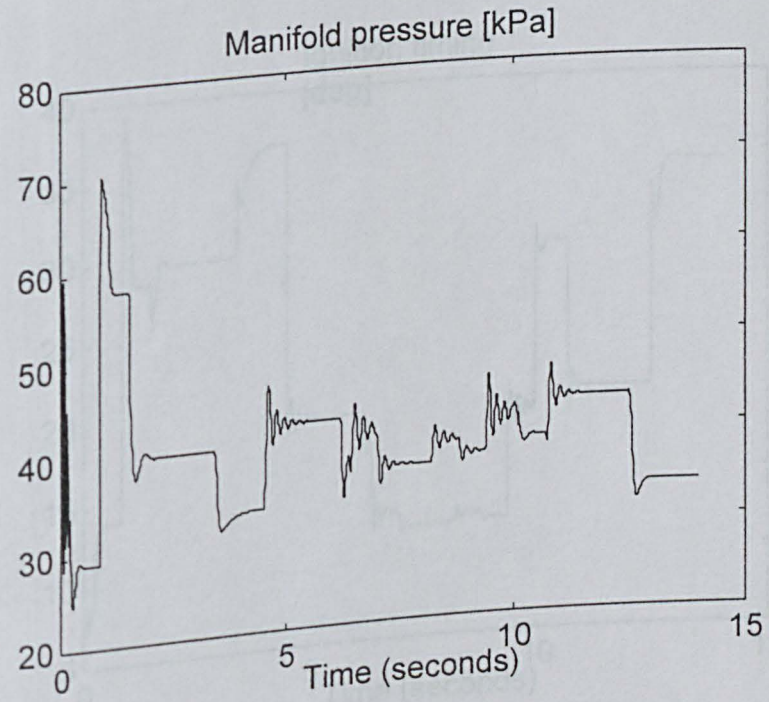


Figure 5.5b Manifold pressure response (under load)

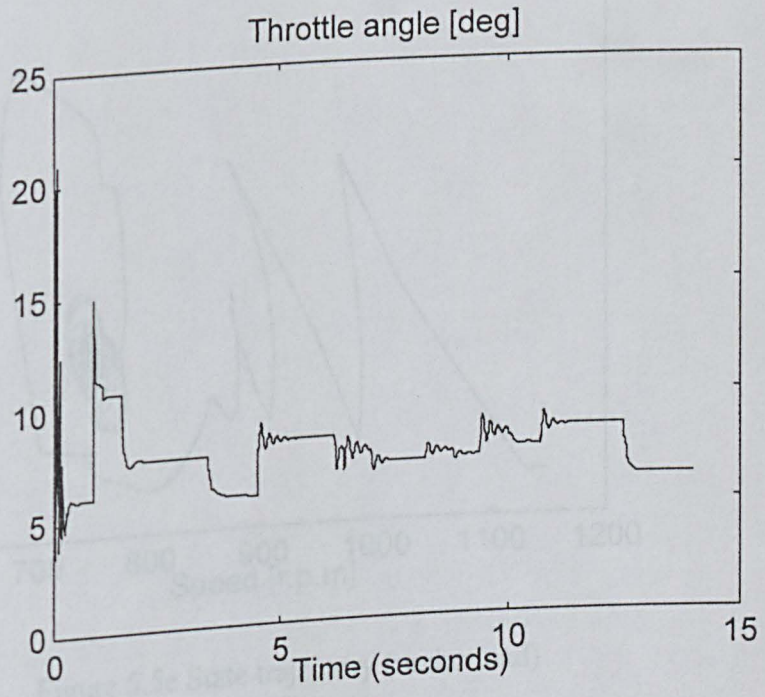


Figure 5.5c Throttle angle (under load)

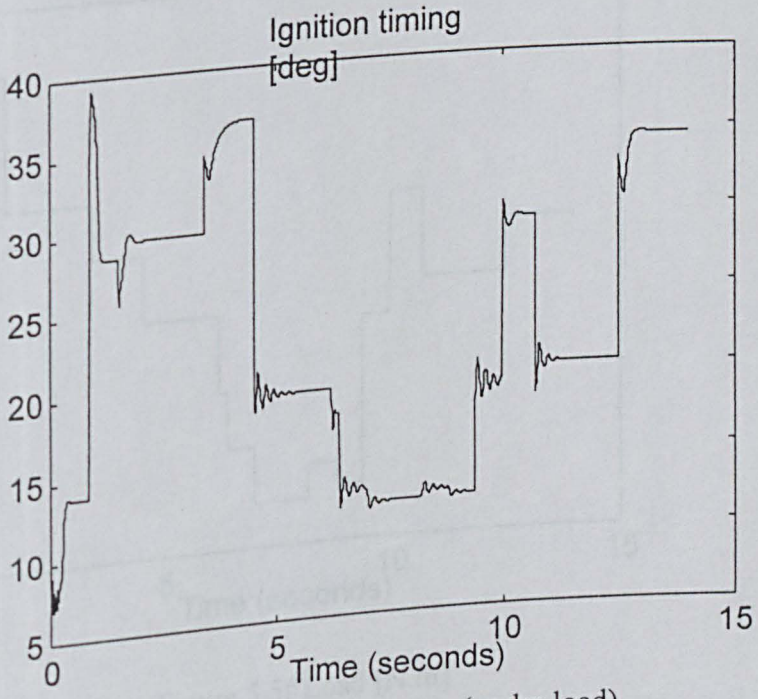


Figure 5.5d Ignition timing (under load)

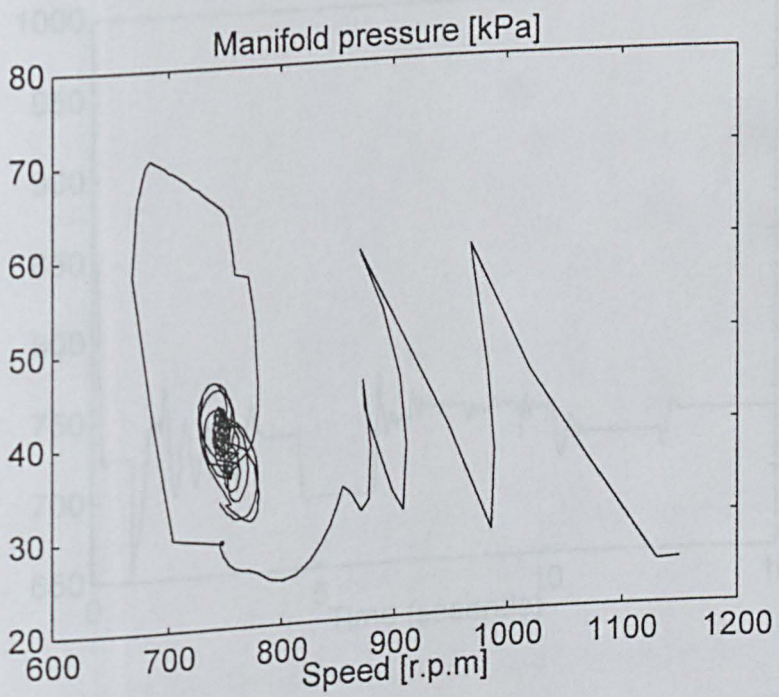


Figure 5.5e State trajectory (under load)

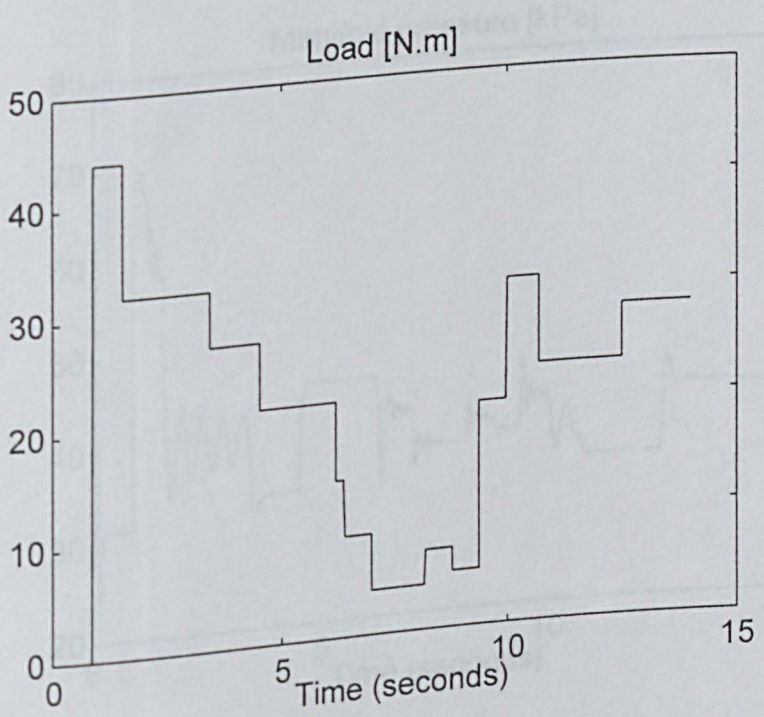


Figure 5.5f Load [N.m]

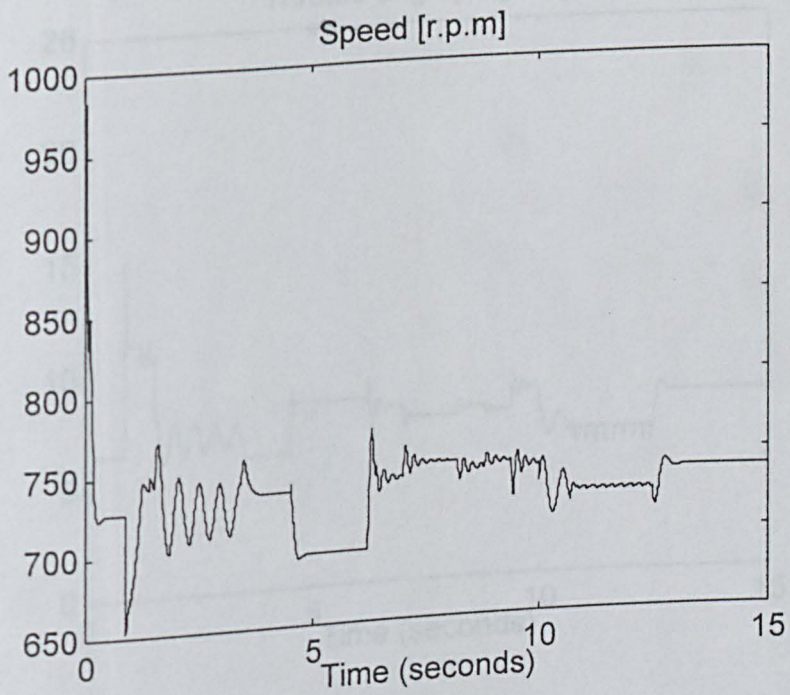


Figure 5.6a Speed response (under load (robustness test))

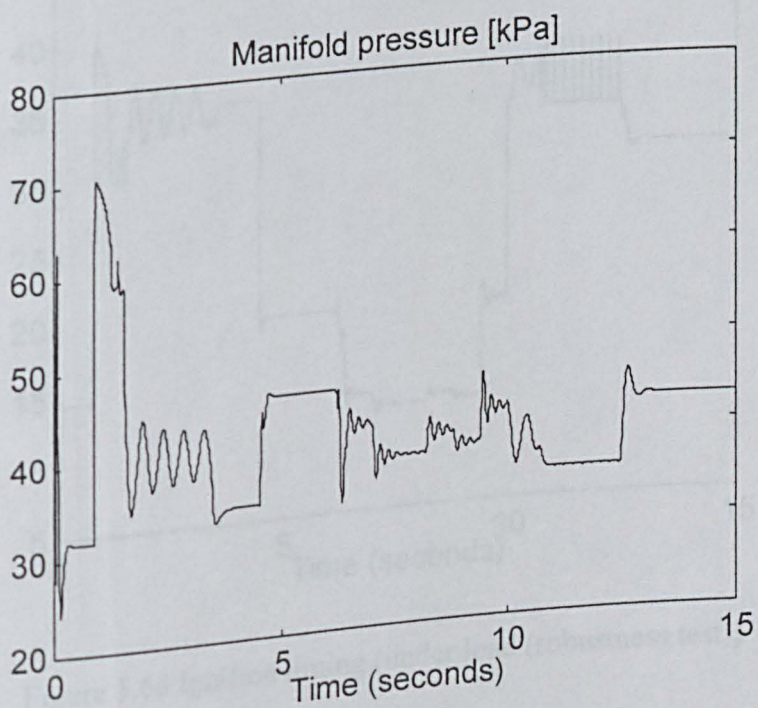


Figure 5.6b Manifold pressure response (under load (robustness test))

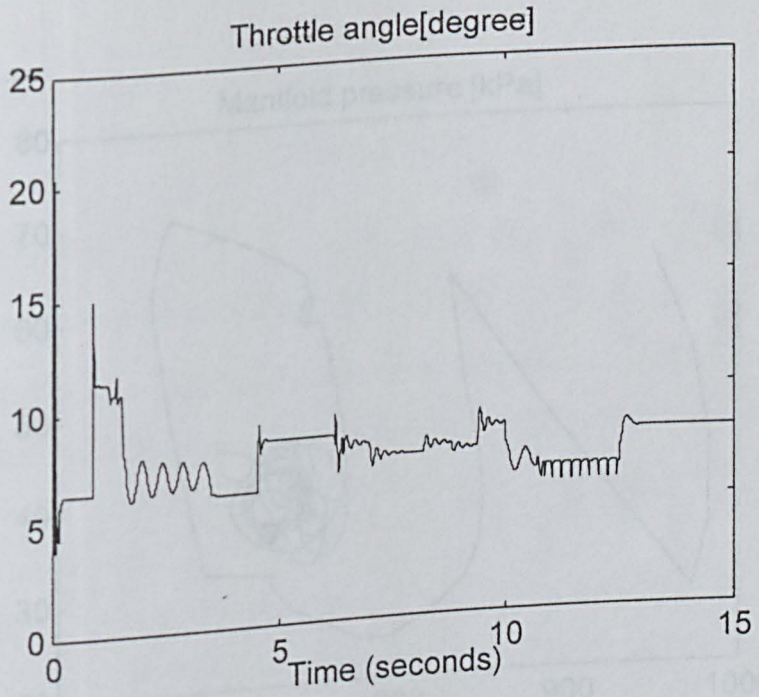


Figure 5.6c Throttle angle (under load (robustness test))

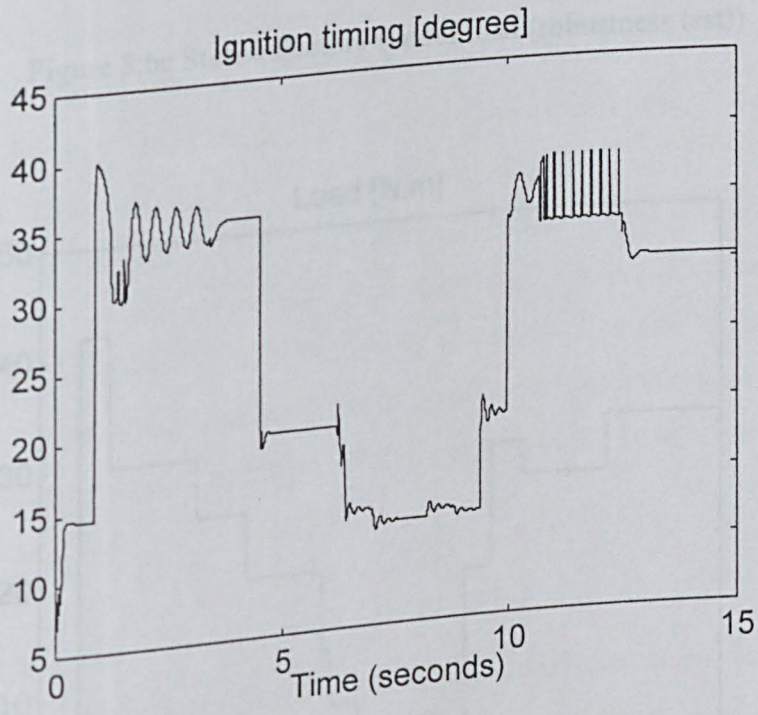


Figure 5.6d Ignition timing (under load (robustness test))

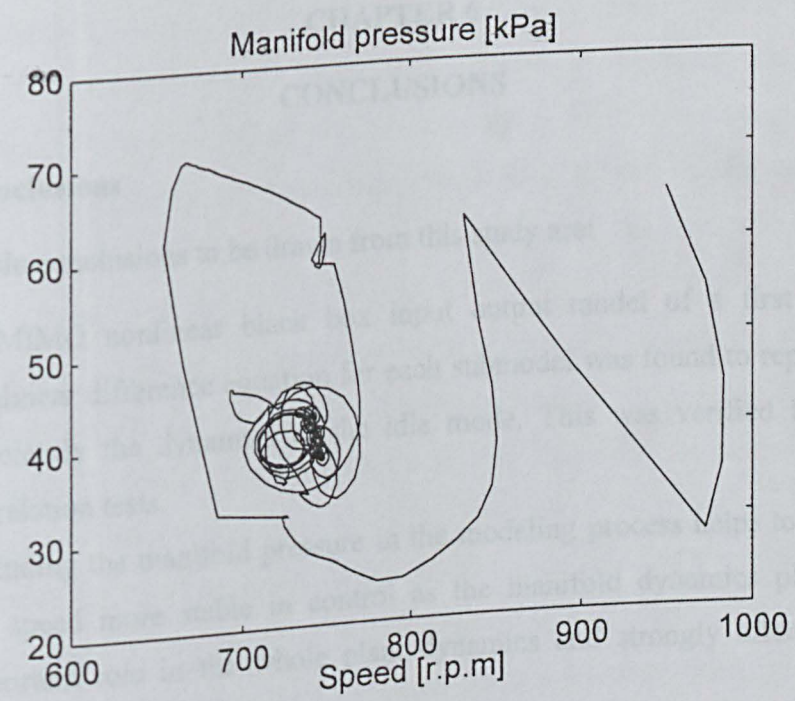


Figure 5.6e State trajectory (under load (robustness test))

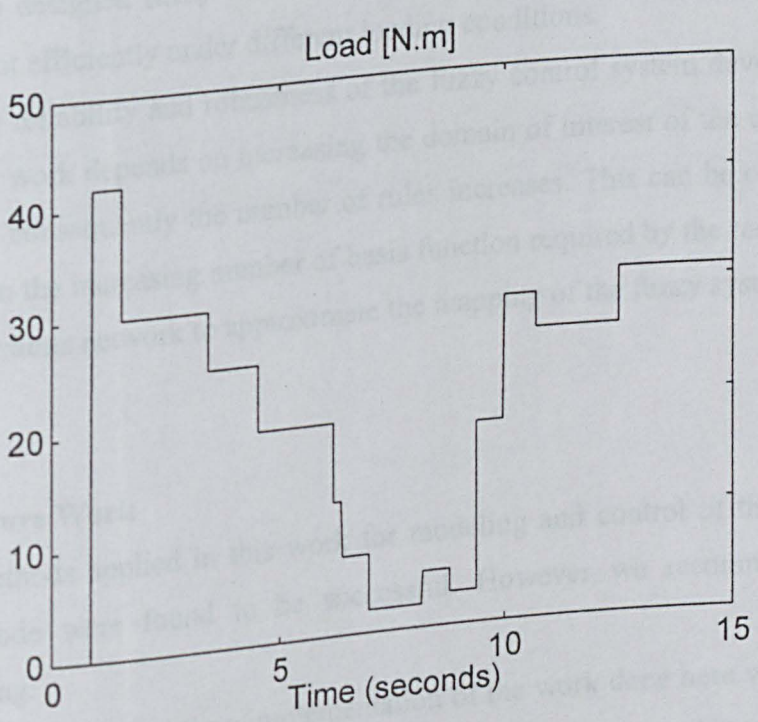


Figure 5.6f Load (under load (robustness test))

CHAPTER 6

CONCLUSIONS

6.1 Conclusions

The main conclusions to be drawn from this study are:

- A MIMO nonlinear black box input output model of a first order nonlinear difference equation for each submodel was found to represent efficiently the dynamics of the idle mode. This was verified by the correlation tests.
- Including the manifold pressure in the modeling process helps to make the speed more stable in control as the manifold dynamics play an important role in the whole plant dynamics and strongly affects the speed dynamics.
- The identification experiment must be run carefully to emulate the real time operating condition of the engine in every day life.
- The designed fuzzy controller can drive the system to its equilibrium point efficiently under different loading conditions.
- The reliability and robustness of the fuzzy control system developed in this work depends on increasing the domain of interest of the cell space and consequently the number of rules increases. This can be concluded from the increasing number of basis function required by the radial basis functions network to approximate the mapping of the fuzzy system.

6.2 Future Work

The methods applied in this work for modeling and control of the engine idle mode were found to be successful. However we recommend the following:

- A successful real time implementation of the work done here will be an avenue to extend the project to include other engine control modes like e fuel injection and the cruise control.

- Different operating and maintenance conditions, different users and the maintenance individual's skills, and weathers in different places can affect the aging conditions of the engine and make drastic changes to the engine parameters. In cases like that it is preferred to use adaptive control strategies to compensate for the aging of the engine parts and unit to unit variability.
- A fuzzy controller is some sort of an associative memory, which is called "FAM". The ability of an associative memory to enlarge different system changes is limited unless it is self organizing or adaptive. As mentioned before the system being controlled may suffers drastic changes in its parameters, which could make the FAM performance unsatisfactory. An adaptive fuzzy controller could be an interesting solution, but this solution requires a model in a feedback linearized form (a linear in control form) [Wang (1994), Vandegrift et al (1995), Jagannthan and Lewis (1996), Ying and Lewis (1994)]. That depends mainly on the mathematical form of the state equation of the model (Slotine (1991)) and the possibility of making the suitable state transformations to the mathematical form of the plant model. An interesting solution to this problem will be to investigate a black box model, which is nonlinear in the system states and linear in control.

REFERENCES LIST IN CHRONOLOGICAL ORDER

Abat, M. and Dosio, N."An Applicaiotn of Fuzzy Logic to Engine Control" *Proc of the Fuzzy and Neural Systems and Vehicle Applications 1991 Conference* 1991, Tokyo.

Baumgartner, C.E.,Geering, H.P,Onder, C.H. and Shafal, E. "Robust Multivariable Idle Speed Control", *Proc, of the American Control Conference*, Seattle, June 1986.

Billings, S.A and Voon, W.S.F, " Structure Detection and Model Validity Tests in the identification of Nonlinear Systems", *Proceedings of IEE. Part D,* 1983, pp. 133-199.

Billings, S.A and Voon, W.S.F, " Correlation based Model Validity Tests for Nonlinear Models ", *Int. Jour. of Control* Volume 44, No. 1, 1986, pp. 235-244.

Billings, S.A and Zhu, Q.M, " Nonlinear Models Validation Using correlation Tests", *Int. Jour. of Control* Volume 60, No. 6, 1994, pp. 1107-1120.

Billings, S.A and Zhu, Q.M, " Model Validation Tests for Multivariable Nonlinear Models including neural Networks", *Int. Jour. of Control* Volume 62, No. 4, 1995, pp. 749-766.

Bohlin, T. " Maximum Power Validation of Models Without Higher-Order Fitting", *Automatica* Volume 14,1978, pp. 137-146.

Coats, F.E., Jr. and Fruechte, R.D.(1983)" Dynamic Engine Models for Control Development -Part II : Application to Idle Speed Control " *Int . J. of Vehcicle Design,* Technological Advances in Vehicle design Series, SP4, Application of control Theory in the Automotive Industry,pp.75-88.

Chen, S. and Billings, S.A., " Representation of Nonlinear Systems the NARMAX model", *Int. Jour. of Control* Volume 49, No. 3 ,1989, pp. 1013-1032.

Chen, S. , Billings, S.A., and Luo,W. " Orthogonal Least Square Methods and their Application to Nonlinear System Identification", *Int. Jour. of Control* Volume 50, No. 5, 1989, pp. 1873-1896.

Chen, S. , Billings, S.A., Cowan, C.F.N and Grant, P.M, " Practical Identification of NARMAX models Using Radial Basis Functions", *Int. Jour. of Control* Volume 52, No. 6,1990, pp. 1327-1350.

Chen, S., Cowan, C.F.N and Grant, P.M, " Orthogonal Least Square Learning Algorithm for Radial Basis Function Networks", *IEEE Trans. Neural Networks,* Volume 2, No. 2 , 1991, pp. 302-309.

Chen, S. , Cowan, C.F.N and Grant, P.M, " Orthogonal Least Square Algorithm for Training Multi-Output Radial Basis Function Networks", *IEEE Trans. Neural Networks,* Volume 4 , No. 2 ,1993, pp. 336-339.

Cook, J.A., Grizzle, J.W., Sun, J. "Engine Control", The Control Handbook, *CRC Press*, pp. 1261-1274 (1996)

Delosh, R.G., Brewer, K.J. , Bush,L.H., Ferguson, L.H. and Tobler, W.E. "Dynamic Computer Simulation of a Vehicle with Electronic Engine Control" SAE paper 810447, 1981.

Dobner, D.J.(1983)" Dynamic Engine Models for Control Development -Part I : Nonlinear and Linear Model Formulation " *Int . J. of Vehcicle Design,* Technological Advances in Vehicle design Series, SP4, Application of control Theory in the Automotive Industry,pp.54-74.

Hazell, P.A. and Flower, J.O., "Sample DataTheory Applied to the Modeling and Control Analysis of Compression Ignition Engines " Part I *Int. J. of Control*, vpl13,no3 pp549-562. 1971

Hazell, P.A. and Flower, J.O., "Sample DataTheory Applied to the Modeling and Control Analysis of Compression Ignition Engines " Part II *Int. J. of Control*, vpl13,no3 pp609-623. 1971

Hsu, C.S. "A Theory of Cell to Cell mapping Dynamical Systems" *Jour. of Applied Mechanics.* Volume. 47 No. 4, pp 931-939, Dec 1980.

Hsu, C.S. ., and Guttalu, R.S. "An Unraveling Algorithm for Global Analysis of Dynamical Systems: An Application of Cell to Cell Mapping " *Jour.of Applied Mechanics,* Volume. 47 No. 4, pp 940-948, Dec 1980.

Hsu, C.S., and Guttalu, R.S."A Discrete Method of Optimal Control Based Upon the Cell State Space Concept" *Jour. of Optimization Theory and Application.* Volume. 46 No. 4, pp. 547-569, Aug 1985.

Jagannthan, S. and Lewis, F.L. "Discrete-Time Adaptive Fuzzy Logic Control of Robotic Systems" *Proc. of the IEEE.* International Conference on Robotics and Automation. Minneapolis, Minnesota April 1996.

Johansson, R. " System Modeling and Identification", *Prentice-Hall Inc.*, Englewood Cliffs N.J., 1st edition 1993.

Jonas Sjoberg, Qinghua Zhang, Lennart Ljung, Albert Benvensite, Bernard Dely Pierrre-Yves Glorennec, Hakkan Hjalmarsson and Anatoli Juditsky, "Nonlinear Black Box Modeling in System Identification: A Unified Overview" *Automatica* Volume 31, No.12,1995, pp. 1691-1724.

Johansson, R. " System Modeling and Identification", *Prentice-Hall Inc.*, Englewood Cliffs N.J.,1st., edition 1993.

Lee, C.C. "Fuzzy Logic Control Systems: Fuzzy Logic Controller", Part 1,2 *IEEE Trans. Syst., Man, Cyber.*, SMC-20, 404-435. 1990.

Leontarities, I.J. and Billings, S.A " Model Selection and Validation methods for Nonlinear Systems", *Int. Jour. of Control,* Volume 45, No. 1, 1987, pp. 311-341.

Livshiz, M., Sanvido, D.J.and Stiles, S.D.," Nonlinear Engine model for Idle Speed Control", *Proc. of the 33rd., Conference on Decision and control,* Lake Buena Vista, FL- December 1994.

Ljung , L. and Torkel Glad " Modeling of Dynamic Systems", *Prentice-Hall Inc.*, Englewood Cliffs N.J.,1st edition 1994.

Muller, B. and Reinhardt, J.,"Neural Network, An Introduction", *Springer-Verlag* 1991.

Nerrand, O. , Roussell-Ragot, D., Personnaz, L. and Dreyfus, G. " Neural Networks and Nonlinear Adaptive Filtering: Unifying Concepts and New Algorithms", *Neural Computations,* Volume 5, No. 99,1993, pp. 165-197.

Nerrand, O. , Roussell-Ragot, D. Urbani, D. , Personnaz, L. and Dreyfus, G. " Training Recurrent Networks Why and How?. An Illustration in Dynamical Process Modeling", *IEEE Trans. Neural Networks,* Volume 5, No. 2, March 1994, pp. 178-184.

Prabhakar, R. S.J. Citron, and Goodson, R.J." Optimization of Automobile Engine Fuel Economy and Emissions" ASME paper 75-WA/Aut-19,Dec.1975.

Powell, B.K., and Cook, J.A." Nonlinear Low Frequency Phenomenological Engine modeling and Analysis", *Proceeding of American Control Conference*, Minneapolis, MN, June 1987 pp 311-316.

Puskorius, G.V. and Feldkamp, L.A."Automotive Engine Idle speed Control with Recurrent Neual Networks", *Proceeding of American Control Conference*, San Fransico, California, June 1993 pp 332-340.

Slotine, J.-J.E and W.Li "Applied Nonlinear Control", *Prentice-Hall Inc.*, Englewood Cliffs N.J.,1st edition 1991.

Smith, S.M., Nokleby, B.R. and Comer, D.J. "A Computational Approach to Fuzzy Logic Controller Design and Analysis Using Cell state Space Methods" In Kandel, A. and Langholz, G. (Eds), *Fuzzy Control Systems,* pp 397-427. CRC Press, Boca Raton, FL.

Takahashi, T., Ueno, T., Yamamoto, A., and Sanbuuichi, H."A Simple Engine Model for Idle Speed Control", *ASME* paper No. 850291 February 1985.

Vandegrift, M.and Lewis, F.L. "Adaptive Fuzzy Logic Control of Discrete Time Dynamical Systems" *Proc. of the IEEE International Symposium on Intelligent Control.* pp395-401. Aug 1995.

Wang, L.X. " Adaptive Fuzzy systems and Control ", *Prentice-Hall Inc.*, Englewood Cliffs N.J.,1st edition 1994.

Williams, S.J., Hrovt, C., Davey, C., Maclay, J., Crevel, W.V. and Chen, L.F. " Idle Speed Control Using an H infinity approach ",*Proc. of the American Control Conference*, Pittsburgh, June 1989.

Ying, H. and Lee C.S.G., "Fuzzy Model Reference Adaptive Control," In review for *IEEE Systems,Man and Cypernetics,* 1994.

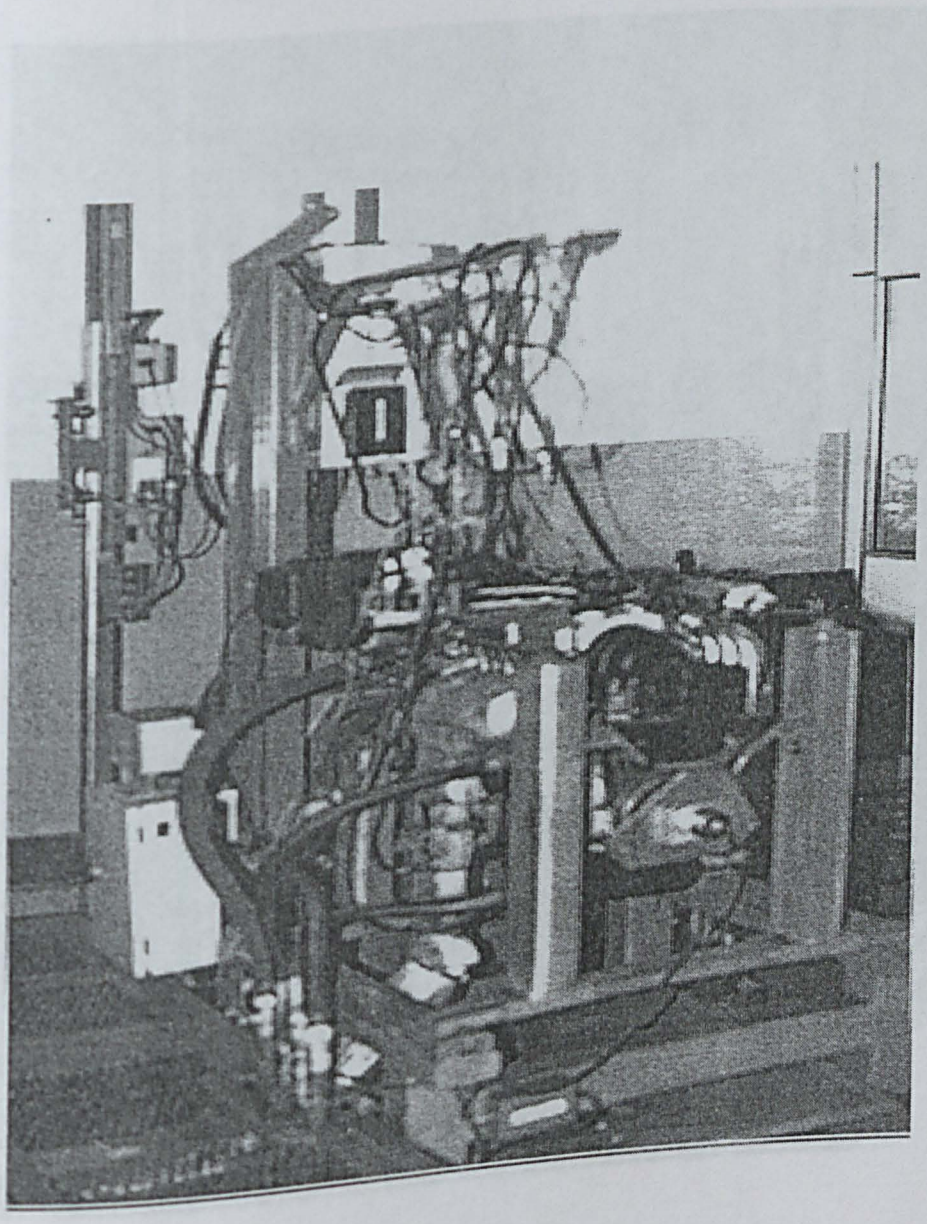


Plate 5.1 the left side of the engine mounted on the engine test bed

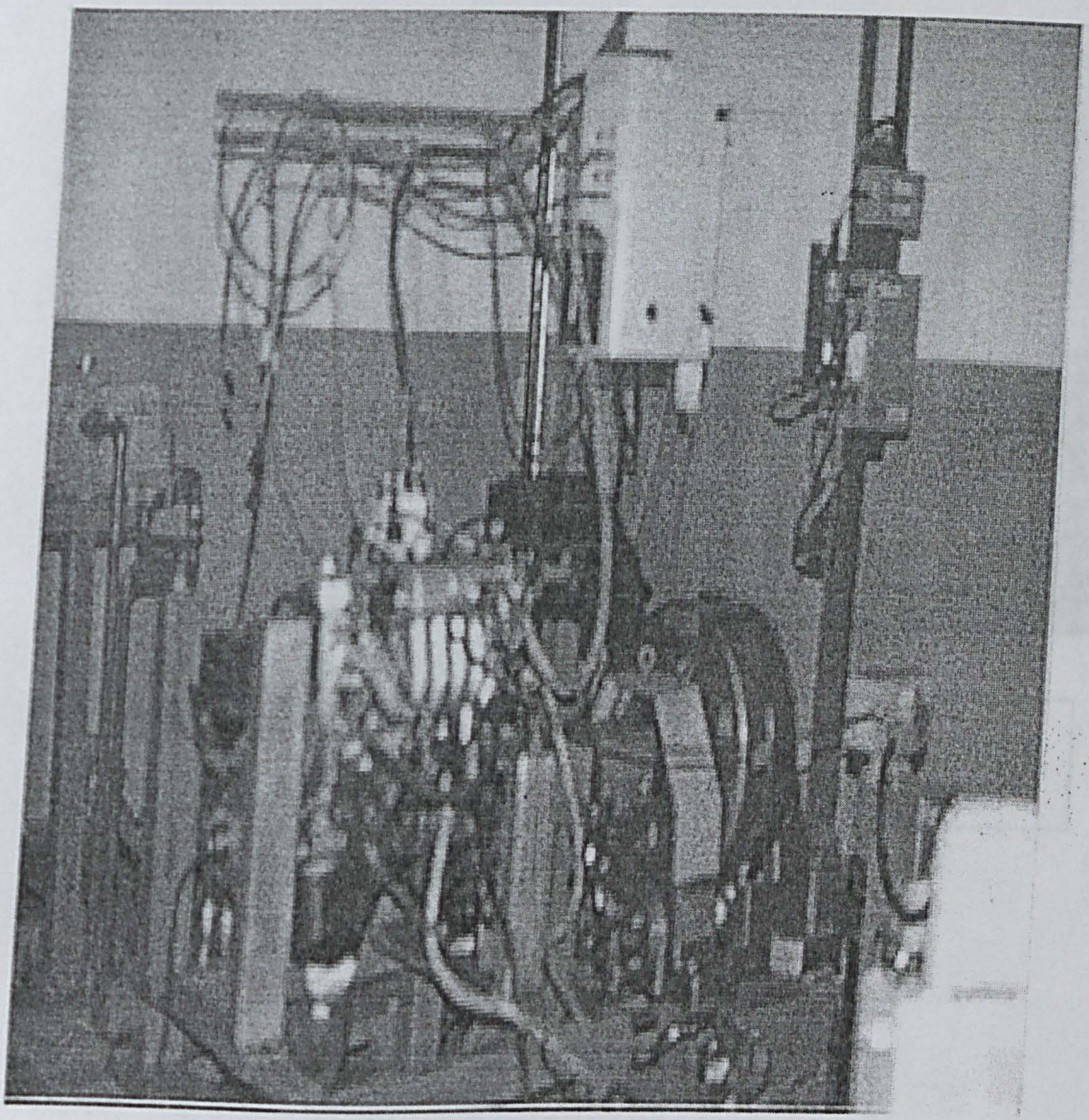


Plate 5.2 the right side of the engine mounted on the engine test bed

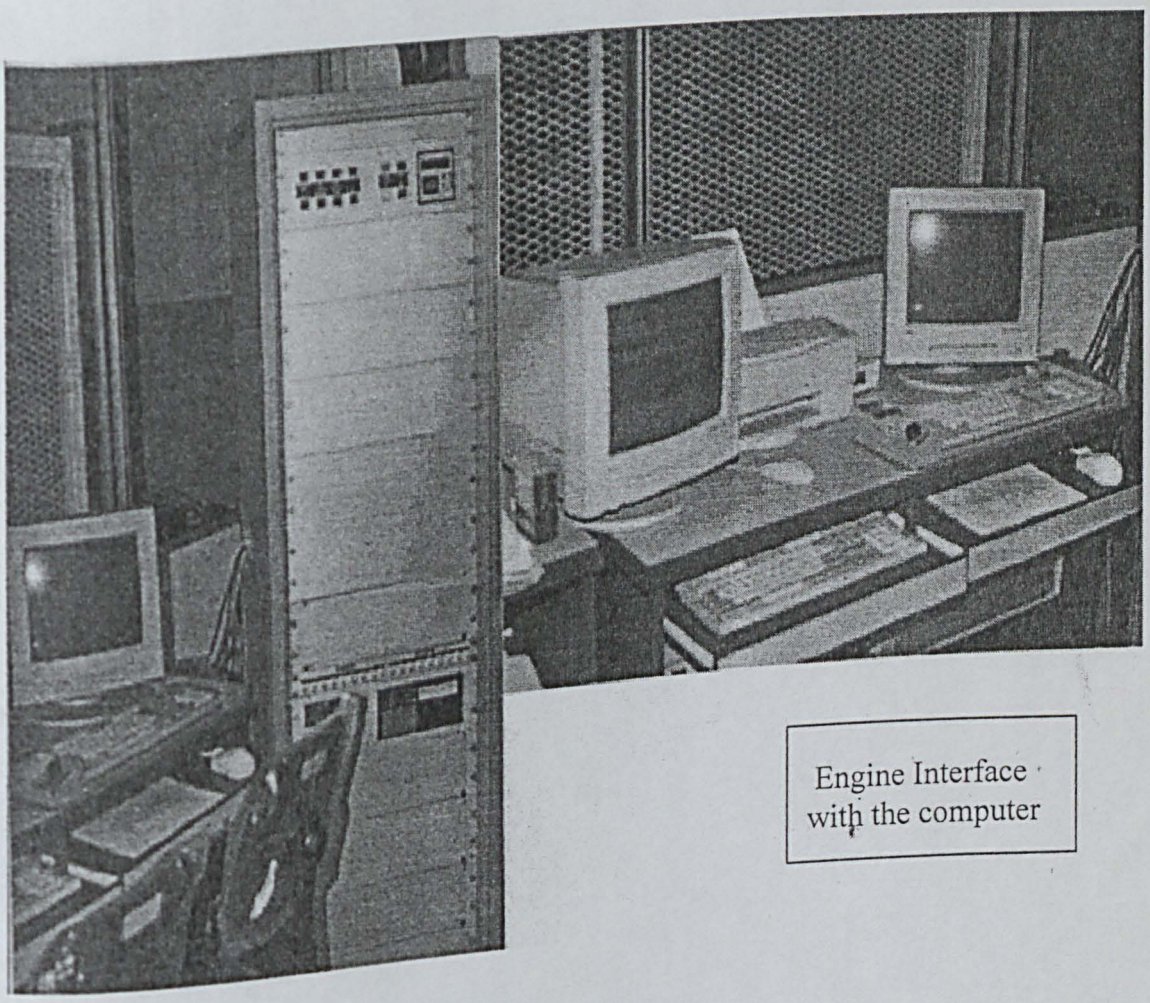


Plate 5.3 the interfacing computer with engine test bed