

**ADAPTIVE CONTROL OF
A SYNCHRONOUS GENERATOR**

BY

PIYASIRI DISSANAYAKA

**A report submitted in fulfilment of the
requirements
for the degree of master of technology
in**

**Department of Electrical & Electrical Engineering
University of Tasmania
Australia**

**Supervisor
Mr. G. The'**

MAY 1993

ABSTRACT

This report presents an investigation of adaptive control of synchronous generator using both a computer simulated model and a laboratory based power generating system. The outcome of the investigation is encapsulated in computer programs supplied with the documentation.

ACKNOWLEDGMENTS

I wish to thank my supervisor Mr. G The' for his suggestions and the clever guidance throughout this investigation. I also like to thank Mr. B Whettett for his support in the laboratory experiments.

I also extend many thanks to my scholarship donor, The Australian International Development Assistance Bureau (AIDAB) for providing financial assistance for me to pursue higher studies at University of Tasmania.

TABLE OF CONTENTS

Page

1. Introduction	1
2. Adaptive control	
2.1 General.	3
2.2 Self oscillating adaptive systems.	5
2.3 Gain scheduling.	6
2.4 Model reference adaptive systems.	7
2.5 Self tuning regulators.	8
2.6 Selection of sampling time	11
2.7 Application of anti-aliasing filter	12
3. Computer simulation of synchronous machine.	
3.1 General.	13
3.2 System equations of 30MW alternator.	15
3.3 Open loop system	16
3.4 System identification	
3.4.1 System model.	18
3.4.2 Time varying parameters.	21
3.4.3 Parameter tracking.	22
4. Controller design and implementation	
4.1 General.	23
4.2 Pole zero placement design.	24
4.3 Design specifications	33
4.4 Simulation	34

5. Laboratory based power generating system	
5.1 Laboratory set up	43
5.2 Adaptive exciter controller	47
5.3 Non-adaptive exciter controller	50
6. Conclusion	53
References	55
Appendices	
1. Simulation model of turboalternator	56
2. Recursive <i>least squares</i> method.	61
3. Runge Kutta method.	64
4. Program listing -Open loop system & plotting.	65
5. Program listing - Adaptive exciter and governor simulation	70
6. Program listing - Laboratory based adaptive exciter	80

CHAPTER 1

INTRODUCTION

Development of computer technology and modern control theory can be linked to achieve better solutions for the control problems. It is possible to propose high performance alternatives for conventional governor and excitation controllers for a synchronous generator. This investigation is based on a design and implementation of an adaptive exciter controller for a synchronous alternator. The study is extended to identify the feasibility of a combined power and exciter adaptive controllers with a computer simulation.

Primary function of the automatic voltage regulator (AVR) of an alternator is to maintain constant operating terminal voltage. In practice this is achieved by adjusting the generator excitation to suit the varying operating conditions of the machine. Alternators in the power system are non-linear systems that are frequently subjected to random load variation with different magnitudes. Fluctuations in the operating conditions eventually make considerable changes in the system dynamics. Hence the conventional fixed gain AVR becomes degraded in performance when the system moves away from the normal operating point. In general, those controllers are tuned around an operating point and operated linearly within limited range. This off tuning can be successfully tackled by the adaptive self tuning controllers.

This report is organised as follows.

In Chapter 2 the basic principals of adaptive control are discussed briefly with various techniques involved. The application of a self tuning controller is exclusively discussed in detail because of it's relevance to this investigation. A computer simulated model for a 37.5MVA turbogenerator is discussed in Chapter

3. The simulation uses fourth order *Runge Kutta integration* to solve non-linear system equations derived using *Park's algorithm* for synchronous machines. The model is tested for its step response by conducting an open loop simulation. Synchronous generator is identified as a plant with a second order discrete transfer function using the *recursive least squares method* with an *Exponential forgetting factor* 0.97.

Chapter 4 explains the design and implementation of the control system using the *pole zero placement* method based on the computer simulated model in Chapter 3. Together with the adaptive exciter controller, a governor controller is implemented at different operating conditions assuming they are independent of each other.

A laboratory based 7.5KVA synchronous alternator driven by a variable speed DC drive is used for the real time simulation. Implementation of the controller and the hardware set up for the experiments are discussed in Chapter 5. The data acquisition system supplied by *BOSTON Technology* is used to interface the computer and the remaining sub systems attached to the machines. Chapter 5 also contains a discussion based on the performance of adaptive and non-adaptive controllers implemented.

As a conclusion, Chapter 6 includes general discussion of the features of controllers by qualifying the experimental and simulated results. Further expansions of the research work conducted and the alternative methods that can be proposed for the controller are also discussed briefly in Chapter 6. The derivation of the plant model, *least squares algorithm* and *Runge Kutta method* are contained in the appendices followed by some of the computer programs used in the experiment.

CHAPTER 2

ADAPTIVE CONTROL

2.1 General

Adaptation means the persistent change of behaviour of a system to cope with new circumstances. In industrial control, adaptive regulators are designed so that they can modify their behaviour in response to changes in the dynamics of the process and the disturbances. At present, ordinary feedback controllers are successfully operated in most of the control systems. The question arises why adaptive control becoming more popular. If it is investigated from a practical view point it is likely that better performances can be achieved.

The concept of adaptive control originated primarily with the aerospace problems. It was found that the classical linear controllers (Fig 2.1) did not always give satisfactory control of the altitude of aircrafts. Response characteristics of these controlled processes varied significantly in flight, and the classical controller could be matched only to a single flight condition. There seemed to be a need for a more intelligent controller which could automatically adjust the changing characteristics of the controlled process. Later it was recognised that such controllers have valuable features applicable for industrial process control.

It is necessary to investigate the distinction between adaptive control and other feedback controls. Ordinary feedback controllers adjust the plant states with reference to fast time scale. Adaptive controllers adjust the plant states with a feedback reference to a slower time scale for updating the regulator parameters. In other words, slowly changing states are viewed as changes in parameters. Regulators with constant parameters are not adaptive because the parameters are independent of the performance of the system.

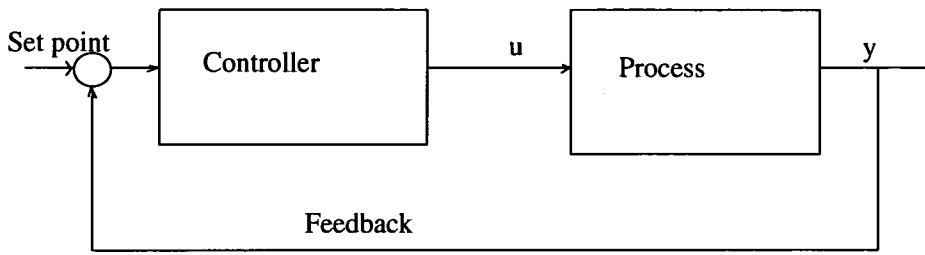


Fig 2.1 Ordinary feedback control system.

It is difficult to decide when adaptive control is useful. There are several reasons why this should be so. One main reason is that all controlled processes for which adaptive control might be suitable are essentially both non-linear and stochastic which is difficult to control and analyse. If they are not non-linear and stochastic there would be no need for adaptation. Non-linear stochastic problems are difficult to control by classical control methods because by definition, there can be no general analytic solutions for them.

It was found that a fixed parameter controller work well in one operating condition that is tuned, but changes in operating conditions may cause difficulties. Changes in the plant may create unstable poles and zeros in the transfer function which may drive the system into unstable situations.

For several decades a wide range of research has been done to implement adaptive controllers using different approaches. Presently, controllers are designed and implemented in discrete time rather than continuous time using latest technologies in sampled data acquisition systems.

There are four different approaches of adaptive control commonly used in the industry. However the latter two methods are the only approaches matching with most definitions of adaptive control.

1. Self - oscillating adaptive systems . (SOASs)
2. Gain Scheduling.
3. Model reference adaptive control . (MRAS)
4. Self tuning regulators. (STRs)

These techniques are discussed in the following sections.

2.2 Self oscillating adaptive systems

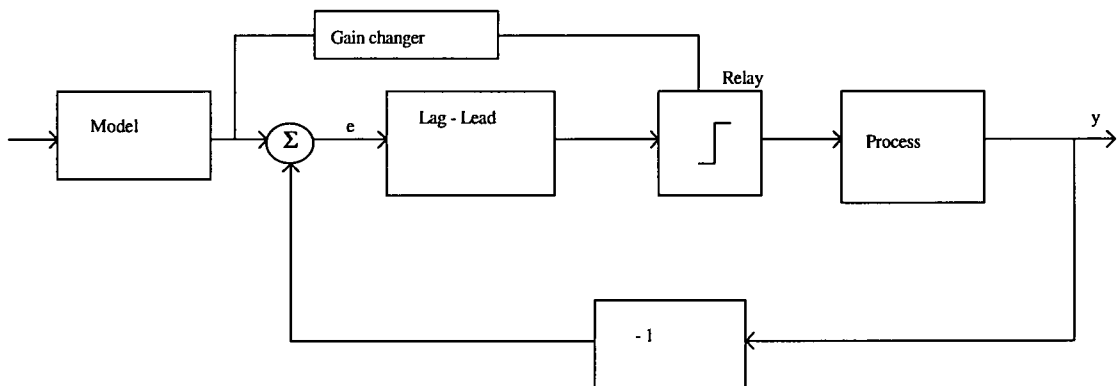


Fig 2.2 Self oscillating adaptive system

The self oscillatory adaptive system (Fig 2.2) is a simple non-linear feedback system that is capable of adapting rapidly to gain variations. This method is based on ideas such as model following, automatic generation of test signals and use of a relay with variable gain. The relay gives high gain for small inputs and the gain decreases with the amplitude of the input signal. The relay often creates limit cycle oscillations in the system. The tracking rate depends on the relay amplitude.

The frequency of oscillation is influenced by a lag-lead compensation network. Such oscillation creates intentional perturbations in the adaptive system, while exciting it all the time. Then the response of the close loop system is relatively insensitive to variations in the process dynamics. The output signal y , follows the reference input over a certain bandwidth defined by the process dynamics. The model gives desired performance for the feedforward path.

2.3 Gain scheduling

Gain scheduling (Fig 2.3) is viewed as a feedback control system in which the feedback gains are adjusted by means of a feedforward compensation. It is rather an open loop adaptation, by monitoring the operating conditions. Selection of the scheduling variables is based on the knowledge of the physics of the system. There is no feedback from the performance of the close loop system that compensates for an incorrect schedule. It would be rather difficult to find the scheduling variables which reflect the operating conditions of the plant. The inputs to the gain schedule are some of the auxiliary measurements of the plant. When scheduling variables are known, the parameters are calculated for different operating conditions.

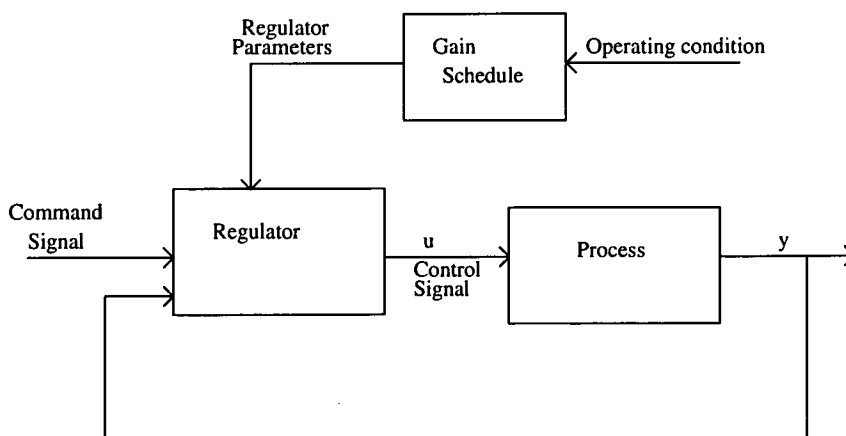


Fig 2.3 Adaptive gain scheduling

The regulator is calibrated for each operating condition and the performance of each condition is checked by simulations. This method has advantages in some cases because the regulator parameters can be changed very quickly in response to process changes.

2.4 Model reference adaptive systems (MRAS)

Model reference adaptive systems (Fig 2.4) were originally developed to design controllers in which the specifications are given as a reference model . The reference model tells ideally, how to respond to the command signal. The reference model is in parallel with the plant rather than in series. If the error between the reference signal and the feedback

signal is equal to zero for all command signals, then perfect model following is achieved. In other words, the adjustment mechanism is designed in order to get y and y_m as close as possible. The MRAS consists of two loops as in Fig 2.4 and is originally proposed by Whitaker (1958). The inner loop provides ordinary feedback, whereas the outer loop adjusts the parameters in the inner loop. Typically the inner loop is assumed to be faster than the outer loop.

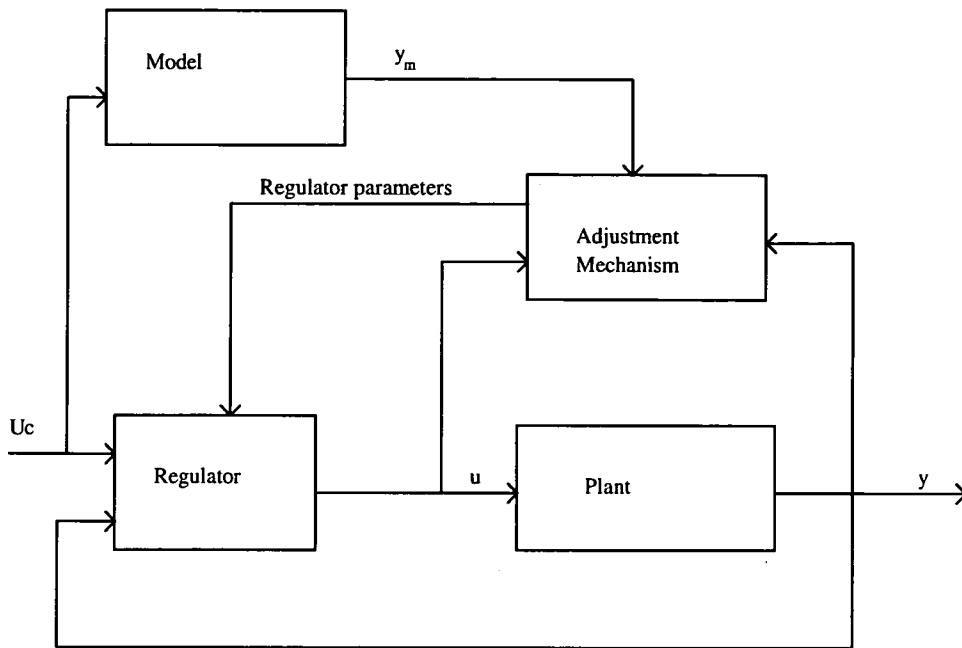


Fig 2.4 Block diagram of a model - reference adaptive system.

2.5 Self tuning regulators (STR s)

All the above adaptive methods are direct methods where, the adjustment rules tell directly how the regulator limit should update. STR is considered as an indirect method where regulator parameters are updated indirectly by parameter estimation and design calculation. Hence MRAS can be treated as a deterministic problem and that the STR is a stochastic control problem. Some of the principals are the identical for both MRAS and STR.

Design of the self tuning controller can be done by two methods. The first self tuning strategy includes system identification followed by controller design and is described as the *explicit* self tuning algorithm. This type of scheme is initiated by Wieslander and Wittenmark (1971) and has the advantage that the self tuner is readily verified by estimated parameters. The second method of STR design discussed by Astrom and wittenmark (1973), and Clark and Gawthrop (1975) is described as *implicit* scheme and briefly discussed later. .

STR with *explicit* identification

The *explicit* self tuner is designed in such a way that the parameters of the plant are estimated separately with on line calculation of controller parameters corresponding to those estimations. Implementing STR with *explicit* identification (Fig 2.5) consists of four steps.

1. Plant parameter estimation scheme to update linearised plant parameters
2. Desired close loop model (Specifications for the design).
3. Controller design procedure.
4. Implementation of control law.

These steps are described in Chapter 4 on Controller Design.

An *explicit* self tuner converges if the parameter estimates converge. This requires that the model structure used in the estimation be correct and the input signal be sufficiently rich in frequencies. Because the *least squares method* is used, the disturbances are not correlated. Also, the control signal is generated from the feedback, and there is no guarantee that it is sufficiently rich in frequencies. For a perfect identification some perturbation signal should be introduced. STR automatically tunes its parameters to obtain the desired properties of the close loop system. In this research work only an *explicit* adaptive self tuning controller is implemented.

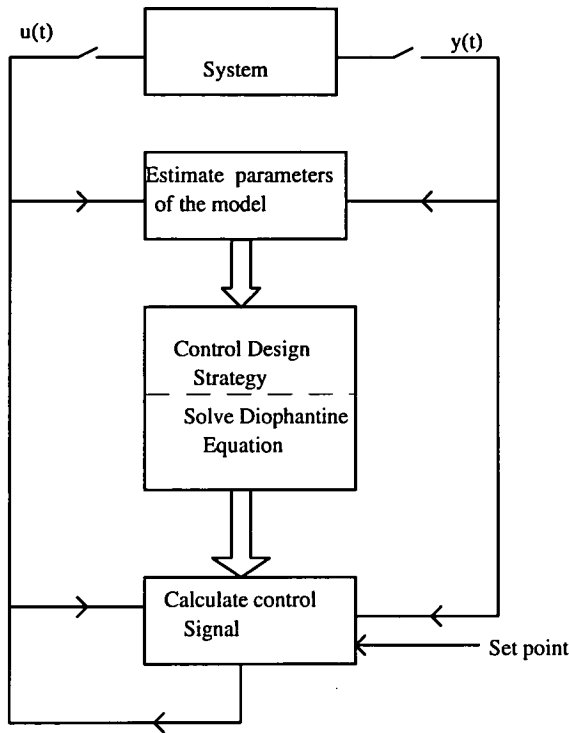


Fig 2.5 Self tuning control with explicit identification.

STR with *implicit* identification

Implicit schemes are first introduced by Astrom and Wittenmark(1973) with an objective of reducing extra computation in identification and control in STRs. In *implicit* schemes, (Fig 2.6) the controller parameters are embedded in the identification procedure, thus reducing the design calculations. Unknown plant parameters are not estimated. Instead, the parameters of the prediction model are used directly to estimate the control signal.

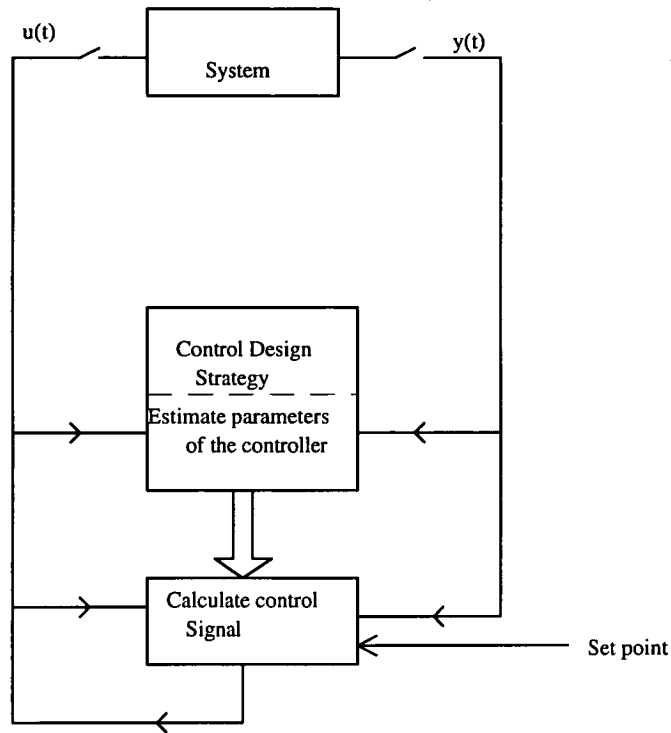


Fig 2.6. Self tuning control with implicit identification.

2.6 Selection of sample time

Selection of sample time is also an important issue in a discrete data system. Too long sampling periods will make it impossible to reconstruct the continuous time signal. Too short sampling time will increase the load on the computer. Sampling time influences properties like; following the command signal; rejection of load disturbances; and measurement noise . As a rule of thumb sampling interval is chosen as ,

$$\omega_0 h = 0.1 - 0.5$$

ω_0 is the natural frequency of the dominant poles of the close loop system. Sampling time is denoted by h .

The choice of the sampling time also determines whether an anti aliasing filter would be taken into account in the design. If the desired crossover frequency of the close loop system is close to the Nyquist frequency then it is not desirable. Increasing sampling time eventually increases the order of the model.

2.7 Application of anti-aliasing filter.

In all digital control applications it is important to have proper filtering of the signals before sampling. It is necessary to eliminate the high frequencies above the Nyquist frequency before sampling. High frequencies may otherwise be interpreted as low frequencies and introduce disturbance in the controller. A filter will make it possible for the estimator to get good models in the correct frequency range. In certain applications, controller D/A output is smoothed by passing it through a post sampling filter. If sampling time is selected such that the cross over frequency is much lower than the Nyquist frequency, then the anti-aliasing filter is not a critical requirement.

CHAPTER 3

COMPUTER SIMULATION OF SYNCHRONOUS MACHINE

3.1 General

The purpose of the simulation is to validate the controller design before adopting it to a real system. To achieve this objective, a hypothetical synchronous machine is modelled on a computer. The machine model has its inputs and outputs as in Fig 3.1

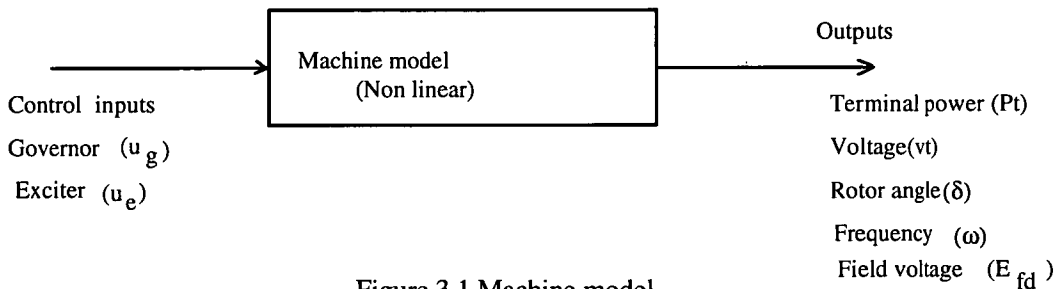


Figure 3.1 Machine model

Simulation of a synchronous alternator is carried out using a machine model based on Park's equations.^[2] The schematic diagram of turboalternator in Fig 3.2 comprises of a synchronous generator, tied via a step-up transformer and a transmission-line to an infinity bus bar. That also includes a representation of the associated prime mover and the field excitation system. A set of differential equations summarised in Appendix 1, defines the system with a 37.5MVA turboalternator with reference to the given system parameters. A synchronous machine is identified as a system described with the following state vectors.

$$X = [\delta, \dot{\delta}, \psi_{fd}, E_{fd}, P_s, T_m]^T \quad (3.1)$$

$$U = [u_e, u_g] \quad (3.2)$$

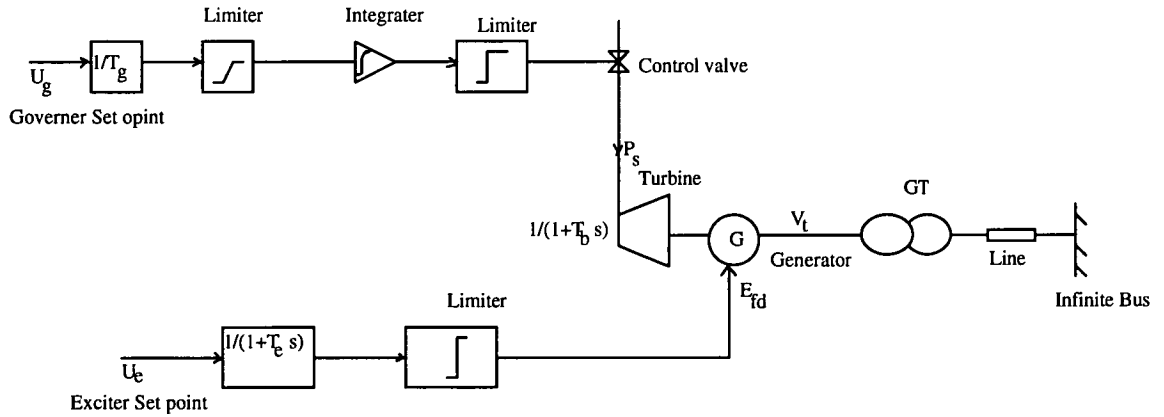
$$Y = [P_t, v_t, \dot{\delta}, E_{fd}]^T \quad (3.3)$$

X = State vector ($X \in \mathbb{R}^6$)

U = Input control vector ($U \in \mathbb{R}^2$)

Y = Output measurement vector ($Y \in \mathbb{R}^4$)

Appendix 1 defines all other state variables mentioned above.



GT = Generator transformer

Figure 3.2 Open loop turbogenerator system

The output vector is chosen in such a way that its elements real power output, machine terminal voltage, rotor angular velocity and field voltage are readily accessible for measurement in a full scale plant. The governor input and the exciter input, which influence the output are chosen as the control variables in the

input control vector. Measurement difficulties exist for some of the variables defined in the state vector. For example, it is difficult to measure rotor angle and field flux linkage. Output predictions can be estimated in terms of input control signals and the past output measurement signals, and it should avoid the reference to any of the state variables. In other words, the plant can be completely defined by the known and measurable vectors rather than unknown state vectors.

3.2 System equations for 30MW Turboalternator

From Appendix 1, The system equations may be written as the following set of first order non-linear equations.

$$\dot{X}_1 = X_2 \quad (3.4)$$

$$\dot{X}_2 = [(X_6 - 1.256 * X_3 * \sin(X_1) + 0.922 * \sin(X_1) * \cos(X_1) - 0.08 * X_2) * 29.637 \quad (3.5)$$

$$\dot{X}_3 = 0.180726 * X_4 - 0.561 * X_3 + 0.422 * \cos(X_1) \quad (3.6)$$

$$\dot{X}_4 = (-X_4 + U_1) * 10 \quad (3.7)$$

$$\dot{X}_5 = (-X_5 + K_v) * 10 \quad (3.8)$$

$$\dot{X}_6 = (-X_6 + X_5) * 2 \quad (3.9)$$

The outputs Y_1 and Y_2 may be expressed in terms of state variables by,

$$Y_1 = 1.256 * X_3 * \sin(X_1) - 0.922 * \sin(X_1) * \cos(X_1) \quad (3.10)$$

$$Y_2 = (v_d^2 + v_q^2)^{1/2} \quad (3.11)$$

where,

$$v_d = 0.798 * \sin(X_1)$$

$$v_q = 0.59 * X_3 + 0.361 * \cos(X_1)$$

It is rather difficult to obtain an analytical solution for the above equations 3.4 to 3.9, to calculate the components of the state vector. The *Runge Kutta* procedure described in Appendix 3, can be used to approximate the solutions of these state equations by selecting the integration step as 0.005 secs.

3.3 Open loop system

The Open loop test is necessary to check the step response of the model with the pre assigned initial steady state conditions. A 12% change in real power from 0.8 p.u. to 0.9 p.u. is considered as the step change to the simulator. The increase in power changes the exciter voltage and the governor input which can be calculated from the steady state conditions derived from state equations. The step inputs to the machine are represented by the new steady state control inputs.

Initial steady state conditions :

$$X_{ss} = [1, 0, 1.152, 2.314, 0.8, 0.8]^T$$

$$Y_{ss} = [0.8, 1.105, 0, 2.314]$$

$$U_{ss} = [2.314, 0.563]$$

With the step increase of the input control vector, $U_{ss} = [2.314, 0.563]$, the system has driven to new steady state conditions.

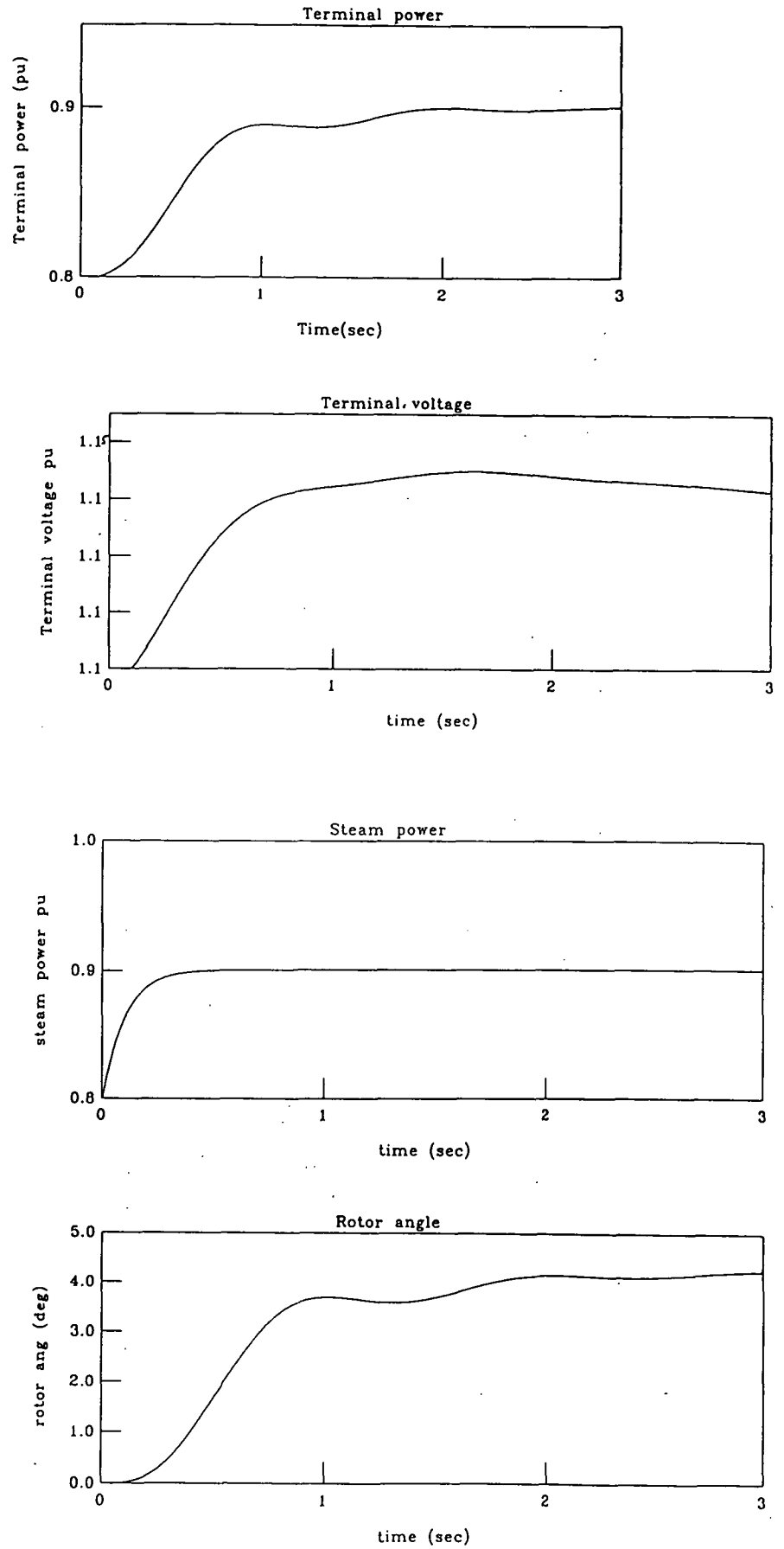


Fig 3.3 Open loop characteristics of the turboalternator

New steady state conditions :

$$U_{ss} = [2.314 , 0.563]$$

$$X_{ss} = [1.078, 0, 1.160 , 2.495 , 0.9 , 0.9]^T$$

$$Y_{ss} = [0.9 , 1.109 , 0 , 2.495]$$

Open loop simulation was conducted using the solutions of the *Runge Kutta* integration formula. Open loop simulation uses the computer program listed in Appendix 4.

Fig 3.3 shows the performance of the open loop system. These open loop results have proved that control laws can be applied to design a stable close loop system.

Fig 3.3.a indicates the terminal power variation against time for the step increase of control inputs. A slower governor creates a significant time lag in terminal power compared to the steam power represented by Fig 3.3.b Fig 3.3.c and Fig 3.3.d which indicate the change in rotor angle and terminal voltage due to the step input.

3.4 System identification

On-line estimation of parameters is one of the key elements in the process of adaptive control and is carried out by applying a prediction algorithm to the unknown system's input and output. Selection of the model structure, experiment design, parameter estimation and validation are the main aspects to be considered with system identification. The *least squares method* is the basic technique applied for this purpose in most adaptive control problems. The technique is much simpler, if the property of being linear in parameters, exist in the selected model.

3.4.1 Model:

The turboalternator system is modelled by a second order discrete transfer function,

$$G(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{Y(z^{-1})}{U(z^{-1})} \quad (3.12)$$

where Y and U refer to the output and input vectors of the system. Parameters b_1, b_2, a_1, a_2 describe the dynamic behaviour of the plant as in the equation 3.12. Obtaining the inverse transform of equation 3.12, it can be represented as,

$$y_k = b_1 u_{k-1} + b_2 u_{k-2} - a_1 y_{k-1} - a_2 y_{k-2} \quad (3.13)$$

Equation 3.13 is a difference equation representing the previous sample values of the input and output that is been used to predict the next output sample. It is a recursive equation, and can be calculated using identified parameters b_1, b_2, a_1, a_2 . A simplified block diagram of the identification process is shown in Fig 3.4.

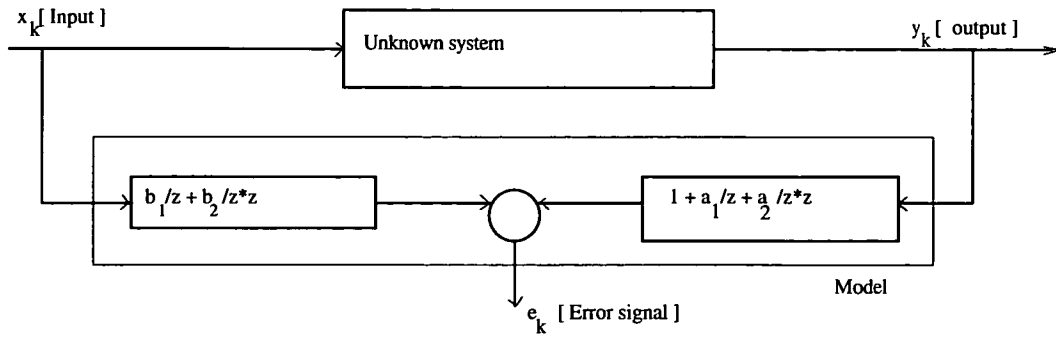


Fig 3.4 System identification process

The error signal generated from the model denoted as e_k is given by,

$$e_k = y_k - b_1 u_{k-1} - b_2 u_{k-2} + a_1 y_{k-1} + a_2 y_{k-2}$$

In matrix form,

$$Y = H\theta + e$$

where H and θ are vectors given by,

$$H = [u_{k-1} \ u_{k-2} \ y_{k-1} \ y_{k-2}]$$

$$\theta^T = [b_1 \ b_2 \ -a_1 \ -a_2]$$

Y is the observed variable vector, whereas H represents known function values consisting of past measurements and control signals that are often termed as regression variables, and θ represents the unknown parameters. The error vector e is to be minimised to obtain the optimal solution for the parameter vector θ . As described in Appendix 2, this matrix equation is solved using the *least squares method* while minimising the squares error represented by the loss function J . The solution for θ is updated in every sample recursively, and implies that the plant parameters are being modified in each sampling interval.

3.4.2 Time varying parameters.

In the least square method, the parameters are assumed to be constant throughout the whole time period. Adaptive control problems are such that the parameters are time varying by nature. The case of parameters that are slowly varying can be simplified to a mathematical model^[1]. It is by replacing least square criterion by

$$J = \frac{1}{2} \sum_1^t \delta^{t-k} (y_k - H^T \theta)^2$$

where the parameter δ lies between 0 and 1, and δ is termed as *Exponential forgetting* or *discounting factor*. The loss function in the above equation implies that the time varying weighting of the data of the data is introduced. The latest data is given a unit weight whereas k time units old data is given a weighting of δ^k . In practice the most suitable value for the *exponential forgetting factor* lies between 0.9 and 1. When $\delta = 1$, the weighting no longer exist in the least square estimation.

To give the late measurements more weight than the earlier measurements an exponential forgetting factor is applied in the formulae A 2.4, A2.5, A2.6 .

It is represented as,

$$\hat{\theta}_{k+1} = \hat{\theta}_k + K_{k+1} (Y_{k+1} - H_{k+1} \hat{\theta}_k) \quad (3.14)$$

and,

$$P_{k+1} = (P_k - K_{k+1} H_{k+1} P_k) / \delta \quad (3.15)$$

where,

$$K_{k+1} = P_k H_{k+1}^T (\delta + H_{k+1} P_k H_{k+1}^T)^{-1} \quad (3.16)$$

The design is crucial with system identification. Excessively fast discounting may cause the parameters to be uncertain and excessively slow discounting will make it impossible to cope with rapidly varying parameters.

3.4.3 Parameter tracking

The ability to track the time varying process parameters effectively is a key issue and a highly desirable property of an adaptive system. The parameter tracking capability depends on the least squares covariance matrix P . Elements of P determine the rate of parameter tracking. These estimators tend to decay to small values rather rapidly. As a consequence, the ability of tracking parameter variation is quickly lost. To improve tracking performance the covariance matrix diagonal elements are to be reset if the trace is very small.

Updating the parameters of the covariance matrix P is possible at different set points, by applying the *Constant trace algorithm*. This can be done by ensuring the trace of P constant at each iteration. The constants of resetting, has a practical value, but is a function of noise and is experimentally selected for the most suitable response.

The plant parameters, i.e., the solution for θ , completely describe the dynamics of the plant at a particular time.

CHAPTER 4

CONTROLLER DESIGN AND IMPLEMENTATION

4.1 General

This chapter is mainly focused on the close loop system that involves controller design based on estimated parameters of the synchronous generator model in Chapter 3. The close loop system should be viewed as an automation of process modelling and design in which the process model and controller are updated at each sampling period.

The synchronous generator modelled in Chapter 3 satisfies the open loop performance of the system that enables it to apply the control law as desired. The implementation of a self tuning controller is based on the *pole zero* placement approach proposed by Astrom and Wittenmark(1980).

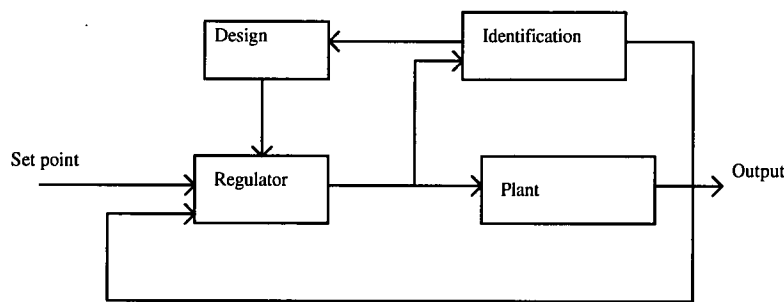


Fig.4.1 Block diagram of SISO(single input single output) Adaptive control system

With the use of the identified plant parameters, the controller can be implemented such that, it matches with the desired close loop characteristics. Out of various methods of designing controllers, pole zero placement design is chosen in this research to implement the controller because it is more straight forward and successful. The main objective is to use a general linear regulator to achieve desired close loop properties.

There exists a difference between tuning and the adaptation process. In tuning, it is assumed that parameters are constant, whereas in adaptation it is assumed that parameters are changing all the time. Therefore to cope with these basic principles in self tuning adaptive problems, it is assumed that the parameters are changing slowly.

4.2 Pole Zero placement Design.

Consider a process with one input u , and one measured output y , which are related by the transfer function, $H(z)$,

$$H(z^{-1}) = z^{-k} \cdot \frac{B(z^{-1})}{A(z^{-1})} \quad (4.1)$$

where $A(z^{-1})$ and $B(z^{-1})$ are polynomials. z^{-1} is backward shift operator.

A and B are relatively prime, i.e., that they do not have any common factors.

Further it is assumed that A is monic, i.e., the coefficient of the highest power in A is unity.

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots$$

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \dots$$

$$k \geq 0$$

The *pole excess* is defined as $d = \deg A - \deg B$, and is the time delay in the process in discrete time systems. Parameters a_1 , a_2 , b_1 and b_2 are estimated by the *least squares* identification. It is desired to find the controller such that the closed loop system is stable and the transfer function of the plant output to the command input is given by,

$$G_m(z^{-1}) = z^{-k} \frac{B_m(z^{-1})}{A_m(z^{-1})} \quad (4.2)$$

where A_m and B_m are co-prime and monic. For stability, the zeros of A_m should be inside the unit circle.

It is not sufficient to specify G_m as the close loop characteristics. With output feedback, there will be additional dynamics that are not excited by the command signal. i.e., the observer dynamics are not controllable from the reference signal. Hence it is necessary to specify the observer dynamics. This is done by specifying the characteristic polynomial A_o as the observer. It influences the sensitivity to load disturbances and measurement.

As discussed by Wittenmark^[1](1980), two main assumptions need to be taken into account in specifying a close loop model. Firstly, it is assumed that the delay in the desired close loop model is at least as long as that in the open loop model. Secondly specifications must be such that unstable or poorly damped process zeros must also be zeros of the desired close loop transfer function.

A general linear regulator proposed by Astrom and Wittenmark^[1], is described as,

$$Ru = T \cdot u_c - S y$$

where u , u_c and y represent the input to the process, controller set point and process output respectively. R , S and T are polynomials in discrete domain.

Now the design is resolved to an algebraic problem of finding polynomials for R , T and S of the adaptive controller.

Fig 4.2 represents the close loop control system with the linear regulator. The close loop equation can be written as,

$$R(z^{-1})U(z^{-1}) = T(z^{-1})U_c(z) - S(z^{-1})Y(z^{-1})$$

Then the close loop transfer function relating y and u_c is given by ,

$$z^{-k} \frac{TB}{AR + z^{-k}BS} \quad (4.3)$$

To carry out the design the polynomial B is factorised as $B = B^+ \cdot B^-$ where B^+ is a monic polynomial whose zeros are stable (inside the unit circle) and so well damped that they can be cancelled by the regulator. When $B^+ = 1$, there is no cancellation of any zeros.

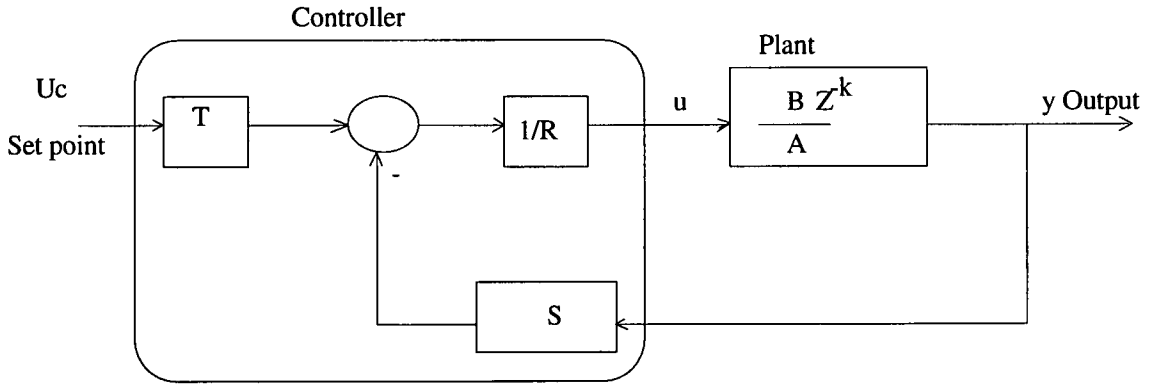


Fig 4.2 Self tuning regulator

To achieve the desired input and output response, the following condition 4.4 must hold.

From equations 4.2 and 4.3,

$$\frac{B T}{A R + B S} = \frac{B_m}{A_m} \quad (4.4)$$

The factors of B that are not also factors of B_m must be factors of R. Factors of B which correspond to close loop zeros, should be cancelled if not desired.

To get a causal controller the close loop model in equation 4.2 must have the same or higher *pole access* as the process in equation 4.1.

This give the condition,

$$\deg A_m - \deg B_m \geq \deg A - \deg B \quad (4.5)$$

Considering the polynomial equalities , the denominator of equation 4.4 becomes,

$$A R + B S = A_m \cdot A_0 B^+ \quad (4.6)$$

Since B^+ is cancelled it is also factors of close loop polynomial ,

$$A R' + B^- S = A_m \cdot A_0 \quad (4.7)$$

Equation 4.6 is called Diophantine equation . Where $R = R' B^+$.

B^+ has been cancelled off .

The numerator of equation 4.4 can be written as,

$$T = A_0 \cdot B_m / B^- \quad (4.8)$$

The solutions of the above equations (4.7) and (4.8) can be justified by considering the constraints for the degree of each polynomial.

The conditions,

$$\deg R \geq \deg T \quad (4.9)$$

$$\deg R \geq \deg S \quad (4.10)$$

ensure that the feedback and feed forward transfer functions are causal. If the time to calculate the control signal in the computer is only a small fraction of the sampling period, then it is acceptable to assume,

$$\deg S = \deg R = \deg T \quad (4.11)$$

If the computation time is close to the sampling period the corresponding relation becomes,

$$\deg R = 1 + \deg T = 1 + \deg S \quad (4.12)$$

This means that there is a time delay in the control law of one sampling period.

From equation (4.5)

$$\deg A_m - \deg B_m \geq \deg A - \deg B$$

$$\deg AR = \deg (AR + BS) = \deg B^+ A_o A_m$$

$$\deg R = \deg A_o + \deg A_m + \deg B^+ - \deg A$$

From 4.8,

$$\deg T = \deg A_o + \deg B_m - \deg B^+$$

It can be shown that, there exists a solution for the Diophantine equation when,

$$\deg S < \deg A$$

Choosing $\deg S = \deg A - 1$, and Since $\deg R \geq \deg S$, from equation 4.9,

$$\deg A_o + \deg A_m + \deg B^+ - \deg A \geq \deg A - 1$$

$$\deg A_o \geq 2 \deg A - \deg A_m - \deg B^+ - 1 \quad (4.13)$$

Equation 4.13 can be used to obtain the degree of the observer polynomial A_o .

This condition implies that the observer polynomial is sufficiently high to ensure the causality of the control law. It should be stable and fulfil the compatibility conditions.

The design is divided in to two distinct cases based on the placement of open loop zeros. The assignment of the desired close loop transfer function is different because in each case, unity steady state gain should be ensured.

Open loop transfer function,

$$H(z) = \frac{B}{A} = K_1 \cdot \frac{z(z+b)}{(z^2 + a_1 z + a_2)} \quad (4.14)$$

where $K_1 = b_1$ and $b = b_2 / b_1$

if $b < 1$ then $B^- = K_1$ and $B^+ = (z+b)$

if $b > 1$ then $B^- = K_1(z+b)$ and $B^+ = 1$

The value of b represents the open loop zero that can lie inside or outside the unit disc. If the zero is inside the unit disc it should be cancelled in the close loop system. If the zero is outside the unit disc, the zero should be included in the close loop because unstable zeros cannot be cancelled. Both cases are separately described for the specific design in this simulation.

Case 1: $b < 1$.

The close loop transfer function is assumed as ,

$$G_m(z) = z \frac{(1 + p_1 + p_2)}{(z^2 + p_1 z + p_2)} = \frac{B_m}{A_m} \quad (4.15)$$

Open loop transfer function is given by ,

$$H(z) = \frac{B}{A} = K_1 \cdot \frac{z(z+b)}{(z^2 + a_1 z + a_2)}$$

where $K_1 = b_1$ and $b = b_2 / b_1$

Because $b < 1$,

$$B^- = K_1$$

$$B^+ = (z+b)$$

By considering the degrees of (4.14) and (4.15),

$\deg A = 2$;

$$\deg B = 2 \quad ;$$

$$\deg A_m = 2 \quad ;$$

$$\deg B_m = 1 \quad ;$$

Hence,

$$\deg S = \deg A - 1 = 1;$$

$$\deg R = 0 \quad ;$$

$$\deg A_o \geq 0;$$

$$\deg T = 1;$$

$$\text{Let, } S = S_0 z + S_1$$

$$T = T_0 z + T_1$$

$$R = R_0 z + R_1$$

Equating the close loop system reference to equation (4.4),

$$\frac{B T}{A R + B S} = \frac{B_m}{A_m}$$

Then ,

$$b_1 (T_0 z + T_1) = (1 + p_1 + p_2) z$$

$$T_1 = 0;$$

$$T_0 = \frac{1 + p_1 + p_2}{b_1}$$

Similarly considering the denominators ,

$$(z^2 + a_1 z + a_2)(R_0 z + R_1) + (S_0 z + S_1)b_1 = (z^2 + p_1 z + p_2)$$

$$R_1 = 0;$$

$$R_0 = 1;$$

$$S_0 = (p_1 - a_1) / b_1;$$

$$S_1 = (p_2 - a_2) / b_1;$$

$$R = (z + b).1$$

The control law can be written as, $RU = TU_c - SY$

$$(z + b)U = T_0 z.U_c - (S_0 z + S_1)Y$$

Converting into a difference equation by taking inverse z transform ,

$$u_k = T_0 u_c - S_0 y_k - S_1 y_{k-1} - b u_{k-1} \quad (4.16)$$

Case 2: $b > 1$.

Since $b > 1$, the zero cannot be cancelled in the desired response. Therefore b should be included in the desired close loop transfer function. To achieve unity gain throughout the control loop, the expression for the transfer function is divided by $(1+b)$.

Then the desired response is given by ,

$$G_m(z) = \frac{(z+b)}{(1+b)} \frac{(1+p_1+p_2)}{(z^2+p_1z+p_2)} = \frac{B_m}{A_m} \quad (4.17)$$

Open loop transfer function in equation 4.14,

$$H(z) = \frac{B}{A} = K_1 \cdot \frac{z(z+b)}{(z^2+a_1z+a_2)}$$

where $K_1 = b_1$ and $b = b_2 / b_1$

Because $b > 1$ then $B^- = K_1(z+b)$ and $B^+ = 1$

By considering the degrees of (4.14) and (4.15),

$\deg A = 2$;

$\deg B = 2$;

$\deg A_m = 2$;

$\deg B_m = 1$;

Hence,

$\deg S = \deg A - 1 = 1$;

$\deg R = 1$;

$\deg A_o \geq 0$;

Let , $\deg A_o = 1$;

$$\deg T = 1;$$

$$\text{Let, } S = S_0 z + S_1$$

$$T = T_0 z + T_1$$

$$R = R_0 z + R_1$$

$$A_0 = z$$

Equating the close loop system reference to equation (4.4),

$$\frac{B T}{A R + B S} = \frac{B_m}{A_m}$$

Then ,

$$b_1 (T_0 z + T_1) = (1 + p_1 + p_2) / (1 + b)$$

$$T_1 = 0;$$

$$T_0 = \frac{1 + p_1 + p_2}{b_1 (1 + b)}$$

Similarly considering the denominator ,

$$(z^2 + a_1 z + a_2)(R_0 z + R_1) + (z + b)(S_0 z + S_1)b_1 = z(z^2 + p_1 z + p_2)$$

$$R_0 = 1;$$

$$R_1 = b - b \cdot \frac{(b^2 - p_1 b + p_2)}{(b^2 - a_1 b + a_2)}$$

$$S_0 = (p_1 - a_1 - R_1) / b_1;$$

$$S_1 = -a_2 R_1 / b_2;$$

$$R = z + R_1$$

The control law can be written as, $RU = TU_c - SY$

$$(z + R_1)U = T_0 z \cdot U_c - (S_0 z + S_1)Y$$

Converting into a difference equation by taking inverse z transform ,

$$u_k = T_0 u_c - S_0 y_k - S_1 y_{k-1} - R_1 u_{k-1} \quad (4.18)$$

Irrespective of the location of the close loop zeros, the control law remains valid, since low frequency steady state gain of the desired close loop system is maintained as unity.

4.3 Design specification :

The specification of the close loop transfer function needs to include the process zeros and poles. Specifications of all poles and zeros for a higher order system require more parameters. It rarely makes sense in practice to give so much data. It is more desirable to give some global characteristics, such as dominant poles in desired dynamics.

A discrete time system can be specified only to include the dominant poles by selecting A_m as,

$$A_m = 1 - 2.e^{-\zeta.\omega.h}.\cos(\omega.h\sqrt{1-\zeta^2}).z^{-1} + e^{-2\zeta.\omega.h}.z^{-2} \quad (4.19)$$

$$= (z^2 + p_1z + p_2)z^{-2}$$

where,

ζ = Damping coefficient.

h = Sampling interval

ω = Resonance Frequency.

p_1, p_2 are the parameters defined according to equation 4.19.

Equation (4.19) corresponds to a second order continuous time process with damping ζ and frequency of resonance ω sampled with period h . It is often easy to select ζ and ω such that the system gets desired properties. The damping ratio is often chosen in the interval 0.5 - 0.8. The resonance frequency ω , is chosen by considering the required rise time and the solution time.

4.4 Simulation

The Simulation uses the *least squares* identification and *Runge Kutta integration* from the previous Chapter and the designed controller for given specifications. In the implementation of the exciter controller, the exciter close loop is assumed as having following specifications.

Resonance frequency : 4 Rad/s
Damping ratio : 0.7
sampling time : 0.05 Second

Similarly, in the implementation of governor controller, governor close loop is assumed as having following specifications.

Resonance frequency : 1 Rad/s
Damping ratio : 0.9
sampling time : 0.05 Second

Simulations are conducted at four distinct system conditions.

1. Adaptive exciter control with fixed governor setting.
2. Non adaptive exciter control with fixed governor setting
3. Adaptive exciter controller with adaptive governor controller.
4. Adaptive exciter controller with non adaptive governor controller.

Different voltage set points are appropriately set to identify the variation of the output conditions as in the Table 4.1. The simulation was performed for 600 sample values equivalent to 30 seconds. The vector notation used in simulations can be listed as follows.

Time(second)	0 -5	5-10	10-15	15-20	20-25	25-30
voltage set point(p.u.)	1.15	1.22	1.22	1.2	1.33	1.4

Table 4.1

In cases where power controllers are implemented, different power levels are set as in Table 4.2.

Time (Second)	0-12.5	12.5-22.5	22.5-30
Power set point (p.u.)	0.7	0.8	0.95

Table 4.2

Input control vector

$$U = [u1 \quad u2]$$

where $u1$ = Exciter control input

$u2$ = Governor control input

Output measurement vector

$$Y = [y1 \quad y2]$$

where $y1$ = Terminal power

y_2 = Terminal voltage.

U_{sp} = Terminal voltage set point

P_{sp} = Terminal power set point.

Start up Procedure

There are several ways of initialising a self tuning controller based on a priori information about the process. However, a practical value is more feasible. In these simulations each parameter is initialised as one at the starting condition. The initial value for the covariance matrix is chosen as 100 times the unit matrix. These values are not critical because the estimator will reach the proper values within a reasonable time. In practice 10 to 50 samples are more than enough for a good controller to track the plant. A perturbation signal is added at the start to speed up the convergence of the estimator. It is often desirable to limit the control signal to a safe value at the start up.

Case 1 : Adaptive exciter controller with fixed governor setting.

This simulation was conducted to investigate the variation of the terminal voltage at different voltage set points at a fixed governor setting. The terminal voltage is controlled adaptively for the period of 30 seconds, using the simulator program FP_ADV.PAS and the system conditions are recorded through the period. Figures 4.3.a, 4.3.b and 4.3.c and 4.3.d represent results of the simulation obtained from respective variations of the terminal voltage, exciter input and terminal power characteristics. It can be seen that the terminal voltage is capable of tracking the set point with an approximate rise time of 1.5 seconds. Terminal power remains constant throughout the period, except at the transient points where the voltage

set points are changed. The Fig 4.3.c clearly shows how the terminal power behaved with respect to time.

Case 2 : Non adaptive exciter controller with fixed governor setting.

A non adaptive exciter controller was designed from the parameters obtained at a random point of the simulation program for Case 1. The control equation for the non adaptive exciter controller is written as,

$$u1_{k_1} = 1.2423 u_{sp} + 16.022 y2_{k_1} - 15.763 y2_{k_1-1} + 0.1869 u1_{k_1-1}$$

The suffix k1 denotes the present sample value and the suffix k1-1,k1-2 denote the previous sample values.

The terminal power set point was remain unchanged at 0.8 p.u. The terminal voltage set point was changed as in the previous case. The computer program FP_NAV.PAS was used to simulate the controller and the operating conditions were recorded for a period of 30 seconds. Figures 4.4.a, 4.4.b and 4.4.c represent the corresponding variations of system states.

It can be seen that from fig 4.4.a, the parameter tracking is not as accurate as the adaptive controller in Case 1.

Case 3. Adaptive exciter and adaptive governor controller.

From the results of Case 1 it is reasonable to assume that the variation power and terminal voltage are independent of each other except at disturbances. That happens because of the change in the relative rotor position of the generator at disturbance points. But Fig 4.3.c shows that the steady state terminal power doesn't change as long as the governor set point remains constant (Fig 4.3.d) The exciter controller was implemented in a similar manner to the previous adaptive

controller in Case 1. A power controller was also implemented using the same techniques, with a unity *exponential forgetting factor* in the identification. The computer program ADV_AD.PAS (Appendix 4) was used for the simulation and the operating conditions were recorded for the period of 30 seconds. The results are plotted in Figs 4.5.a, 4.5.b, 4.5.c .

From the results it is clear that , the combined adaptive and power controller functions better than previous cases (Case 1 and Case 2). Because the governor is adaptive, the power changes reflected due to the set point change of terminal voltage would cause a transient in the governor control input.(Fig 4.5.d). From figures 4.5.a and 4.5.c proved that the time constant of the governor is comparatively smaller than the time constant of the exciter and the controllers are capable of tracking respective set points.

Case 4. Non adaptive governor and adaptive exciter controller.

The parameters of the governor controller was obtained at a randomly selected point of the simulation in Case 3.

The control equation for the governor control signal was then calculated as ,

$$u2_{k_1} = 0.195 P_{sp} - 1.402 y1_{k_1} + 0.671 y1_{k_1-1} + 0.1762 u2_{k_1-1}$$

The simulation was conducted with set point variations as in the tables, to implement adaptive voltage and non-adaptive power controller. The results monitored for the period of 30 seconds and are plotted on Figs 4.6.a, 4.6.b and 4.6.c .

The results in Case 3 and Case 4 are close to each other compared to the results in Case 1 and Case 2. It can be predicted that the power controller exhibits approximately linear characteristics.

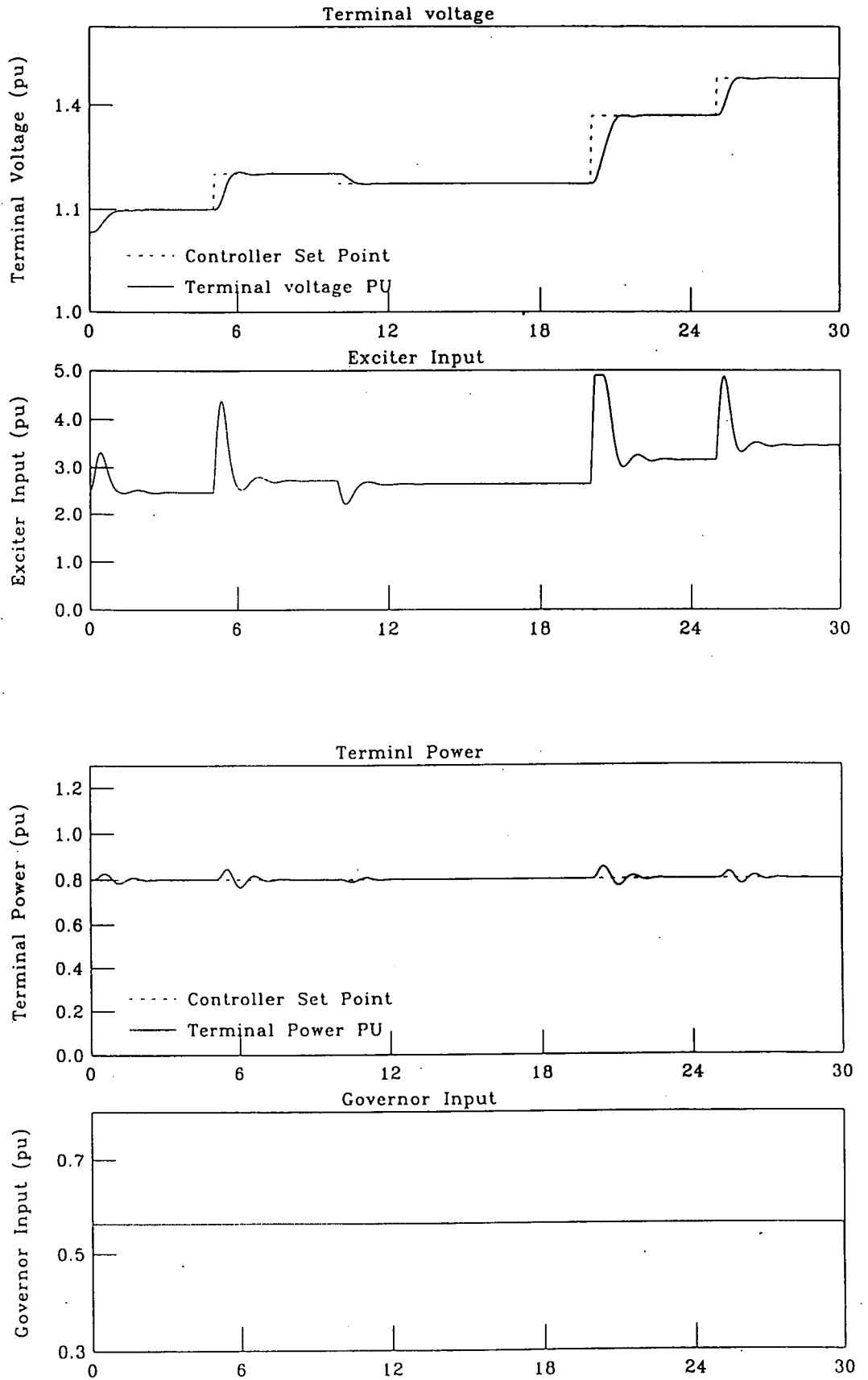


Fig 4.3 Characteristics of adaptive exciter controller with fixed power

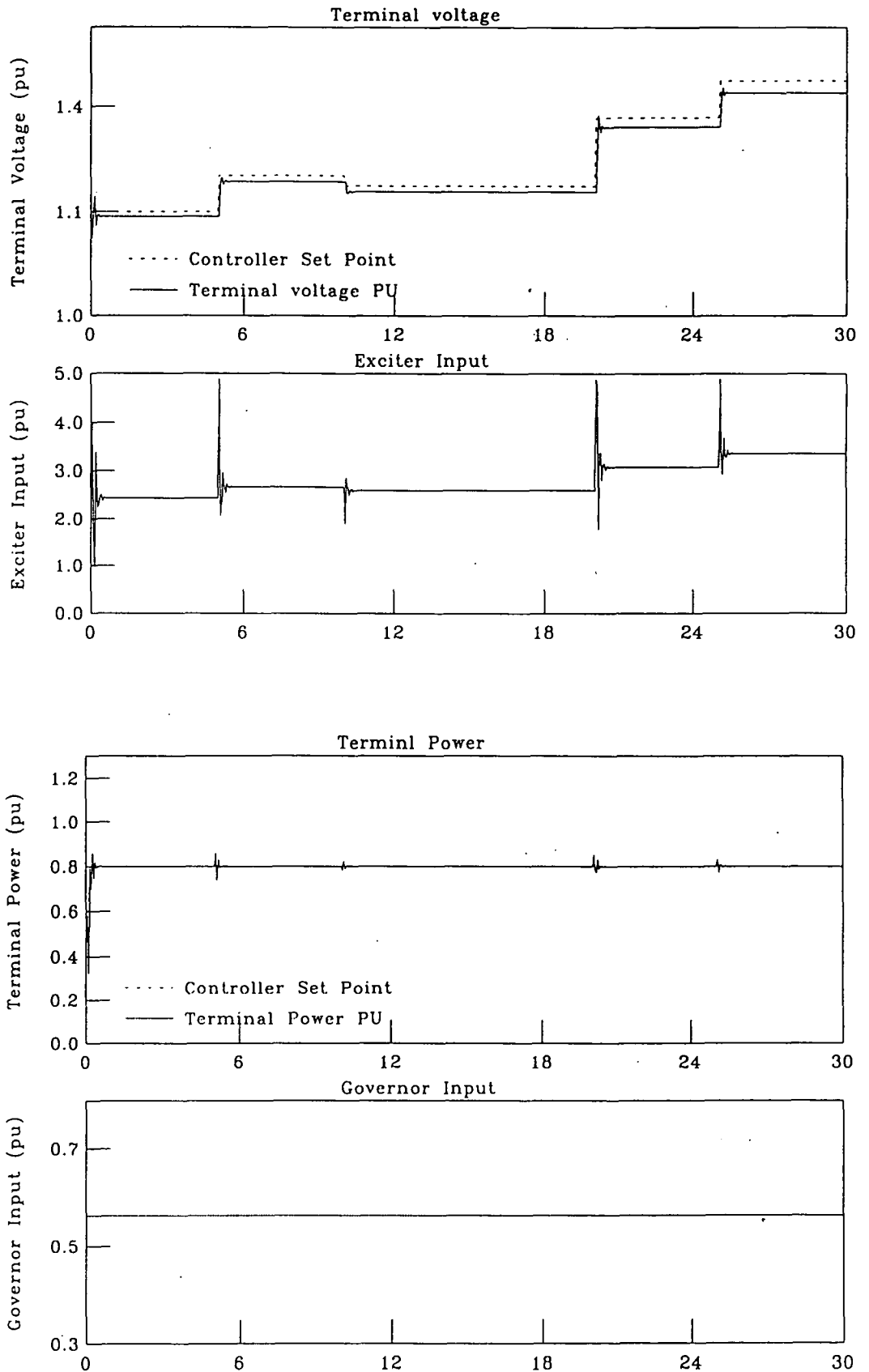


Fig 4.4 Characteristics of Non adaptive exciter controller with fixed power

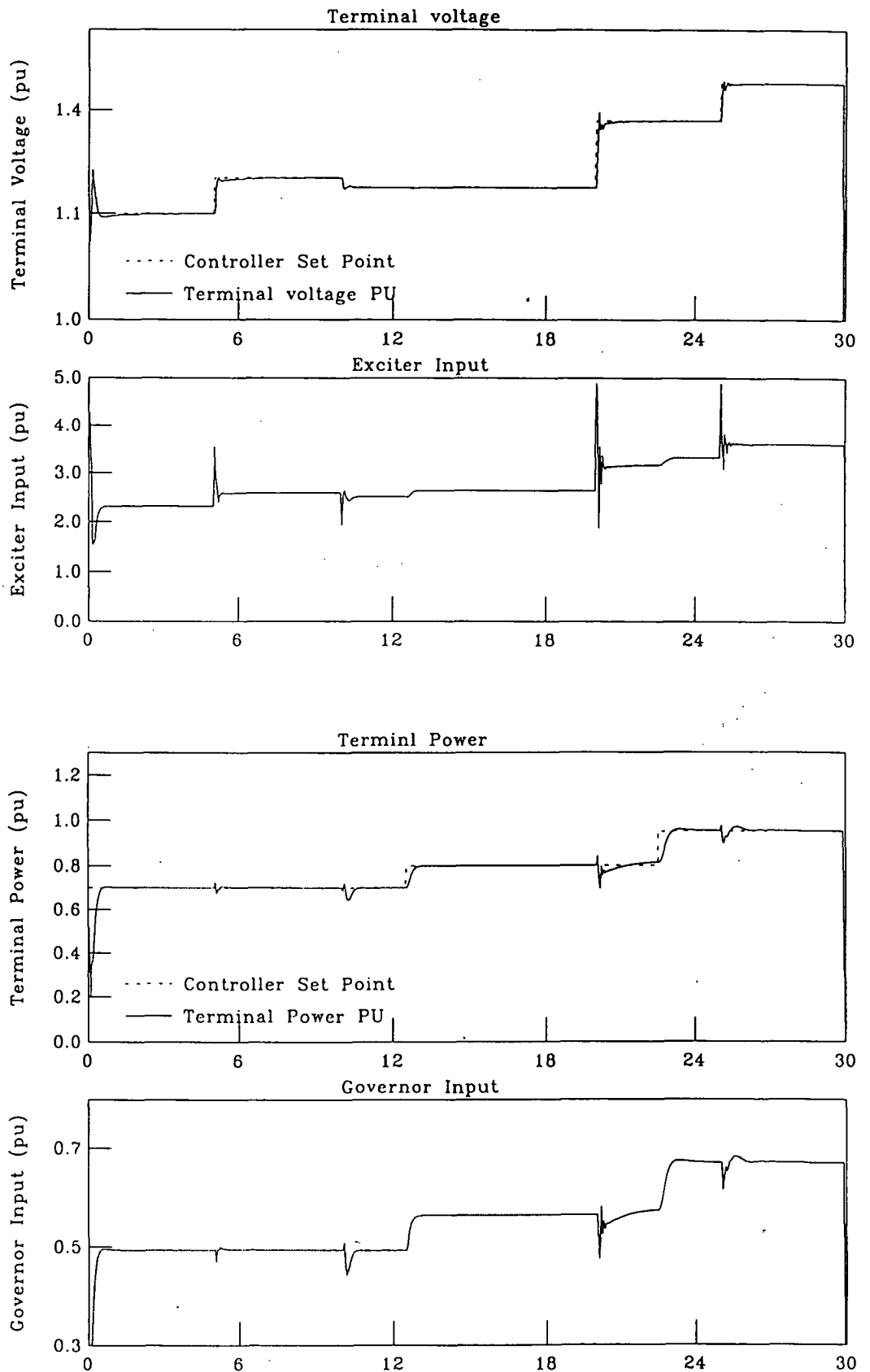


Fig 4.5 Characteristics of adaptive exciter and adaptive governor controller

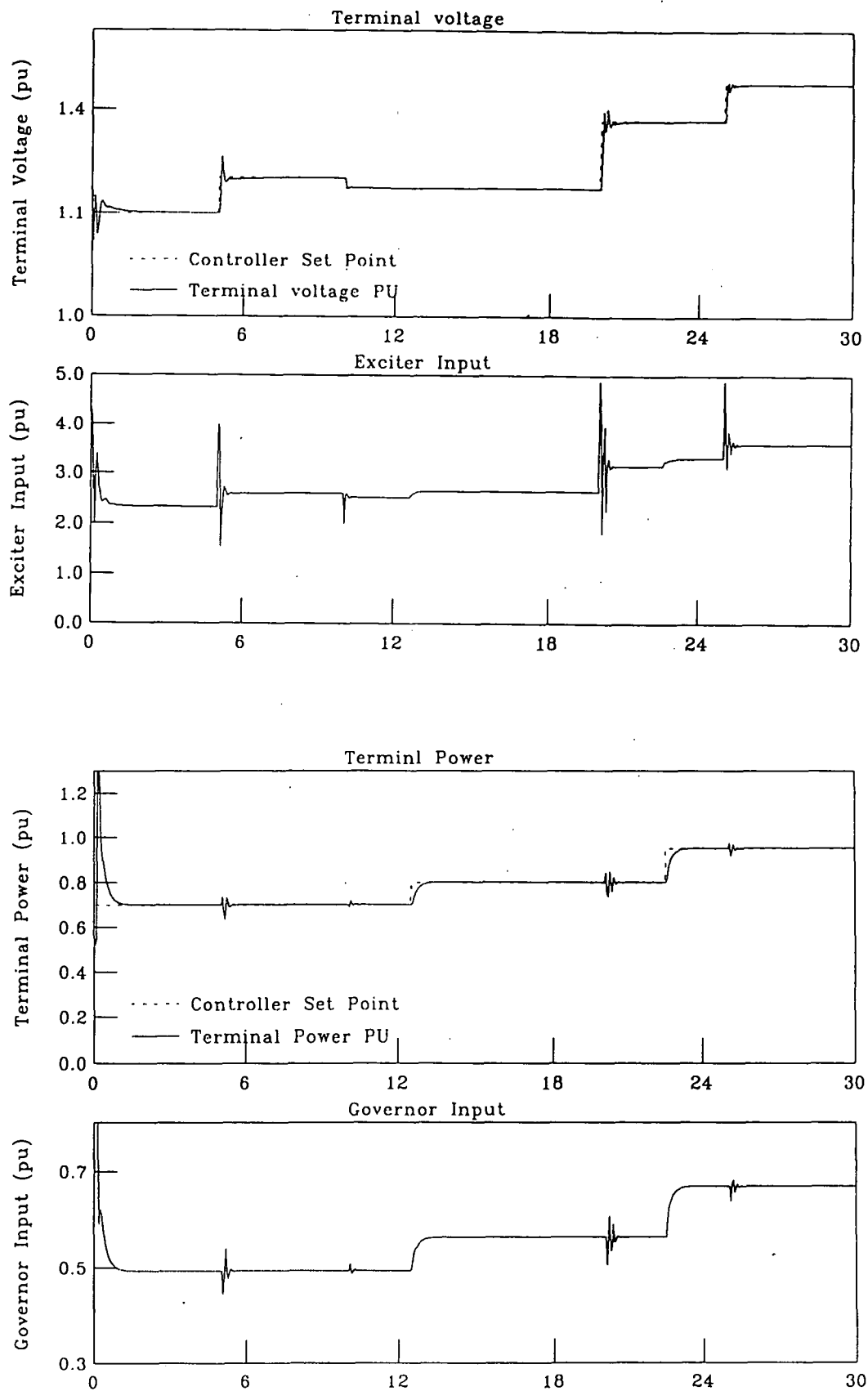


Fig 4.6 Characteristics of adaptive exciter with non adaptive governor

CHAPTER 5.

LABORATORY BASED POWER GENERATING SYSTEM.

5.1 Laboratory set up

The simulations conducted in Chapter 4 are designed for a hypothetical machine model based on Park's equations. The real time power generation systems may have to face more complicated problems in implementations, especially with regard to the limitations of the hardware used in the experiments and the non-linearities in the plant sub systems.

The following block diagram shows the laboratory set up for the power generating system used in the experiments.

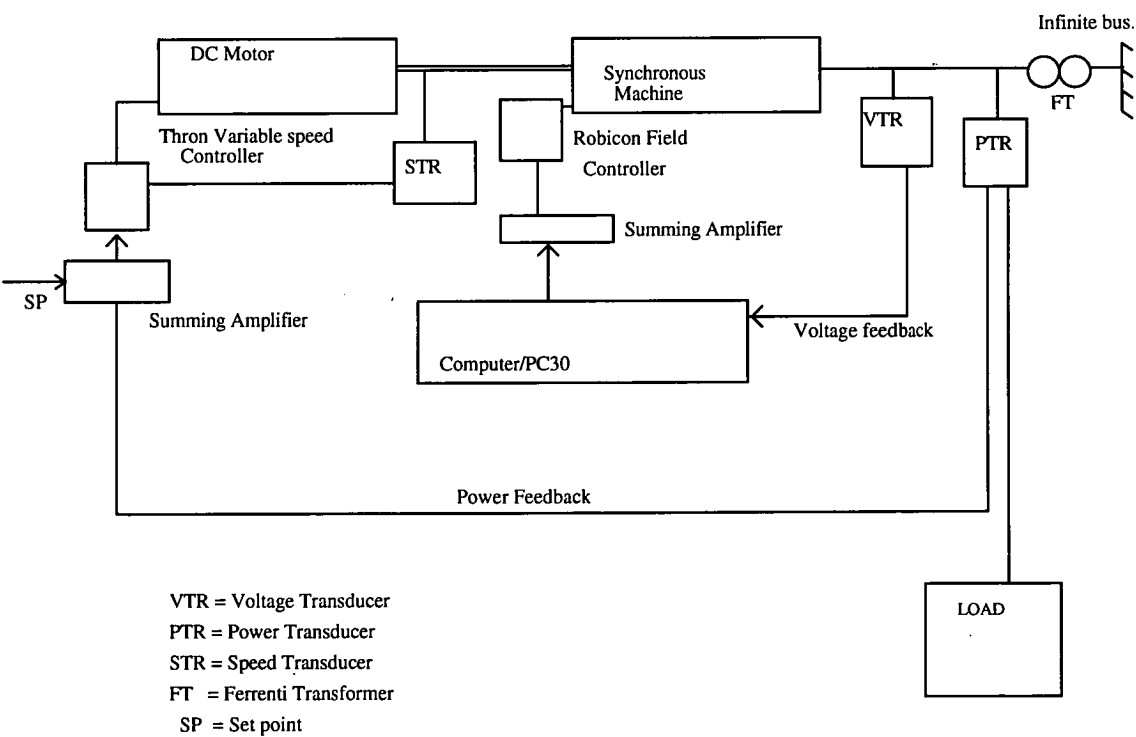


Fig 5.1 Laboratory based system set up.

The laboratory model of the synchronous generator consists of a 7.5 KVA direct coupled motor-generator system. The list of apparatus and hardware needed for the experiment are listed below.

1. A 7.5 KVA synchronous three phase alternator provided with the infinity bus connection as required.
2. A DC motor with output power more than 7.5KVA supplied with fixed field voltage.
3. A thyristor controlled 4 quadrant 380V DC rectifier system, with adjustable output current by gain control.(ROBICON field controller)
4. A thyristor controlled variable speed DC drive, operated with the tacho feedback
and power feed back.(THRON controller)
5. An IBM compatible personal computer installed with PC30 -BOSTON technology
data acquisition system. The real time software (QUINN CURTIS -Turbo Pascal
version) and the Turbo Pascal compiler along with BGI (Borlands Graphics interface) should also be installed in the computer.
6. A Power transducer measure up to 10KW with a DC output of 2V (HOKI)
7. A 12 pulse 3 phase voltage transducer, to measure the terminal voltage of the alternator.
8. An Induction motor load of 5KW (variable) which is wired through a three phase
contactor.
9. Ferrenti transformer in between the infinite bus and the machine terminal.

The complete demonstration of the control system is programmed using TURBO PASCAL , together with other associated software.

The software program SYS2.pas implemented in the personal computer is assigned for the following functions.

1. Construction of the artificial plant parameters according to the least squares estimation.
2. Sensing of operating conditions of the machine and making the corresponding tuning of the controller parameters.
4. Initiating disturbances to the analog simulation by step inputs to the exciter set point.
5. Recording input and output signals and displaying them on the monitor in real time.
6. Recording the operating conditions in a data file.

While changing the voltage set point at different intervals, voltage feedback was passed through the adaptive controller . The input output requirements of the controller are handled by using 12 bit A/D and D/A converters available with the PC 30 module, together with a suitable interface to the alternator. The signal inputs and outputs were sampled with sampling time of 0.1 second, corresponding to the 10 Hz sampling frequency. Specifications for the adaptive exciter controller are as follows.

Resonance frequency : 4 Rad/s

Damping ratio : 0.7

The sampling frequency is more than 10 times higher than the resonance frequency of the exciter control close loop plant. Hence as mentioned in Chapter 2 there is no need to apply an anti-aliasing filter before sampling.

The turbine is represented by the DC motor whose speed is regulated by an electronic turbine governor simulator. The governor simulator is activated by the power feedback and speed feedback from the system. The signal obtained from the power transducer is inverted and summed with a reference signal before being fed in to the governor controller.

The exciter field controller consists of a four quadrant thyristor rectifier which is activated by the signal output from the D/A converter. The D/A signal is summed with a reference DC signal before being fed in to the ROBICON controller. To input the terminal voltage feed back, an A/D channel is selected on the data acquisition module.

The PC 30 module is capable of conducting digital and analog input output operations via the PC bus to the peripherals. It is compatible with the IBM PC, PC/XT, PC/AT, PS2 series of computers. The data acquisition software supplied with PC 30 consists of a set of real time device drivers callable from common languages like PASCAL, C and FORTRAN. 12 bit A/D signal inputs are limited to a full scale of +5 to -5V or 0 to 10V and 12 bit D/A signal outputs are limited to full scale of 0 to 10V or -10 to +10V. Those scales are readily programmable from associated software.

Two main aspects have to be considered by the implementation, namely; the sampling period and the non-linearities in the experimental set up. Saturation of hardware elements, such as transformers in transducer elements and ripple may cause unmodelled dynamics in the system. Since control and the input signals are limited by the range of 12 bit A/D and D/A, they are limited by maximum and minimum amplitude. Also, numerical saturation in the personal computer may cause problems. The averaging type power transducer used in this experiment, does not show the instantaneous power. Hence the response time and the sampling

time may have a certain mismatch. However, in the experiment it was of minor importance.

The variation of terminal voltage is monitored using the real time graphics at each 0.1 second at different operating conditions given below.

1. Starting condition.
2. Synchronising the machine via Ferrenti transformer.
3. Loading the machine with inductive load.
4. Isolating the bus.
5. Unloading the machine.

5.2 Adaptive exciter controller

The experiment was conducted in two phases. Firstly, an adaptive exciter controller was implemented using the computer program REAL.PAS. The experiment was conducted for the period of 130 seconds. The exciter input signal and the terminal voltage were monitored throughout the period and recorded in a data file. The power control loop was operated as a linear feedback to the governor simulator.

Terminal voltage set points are varied as is the Table 5.1. The experimental results are plotted in Fig 5.2.a and 5.2.b.

The results in Fig 5.2 should be carefully examined at the points of forced disturbances. At the 9th second after start, the isolated machine was connected to the infinity bus via the Ferrenti transformer. It acts as an impedance of a step-up transformer and a transmission line section in a full scale power plant.

Time (seconds)	Voltage set point(p.u.) 1 p.u. = 415 V
0-10	0.987
10-20	1.054
20-30	1.085
30-40	0.999
40-50	0.9627
50-60	0.987
60-70	1.054
70-80	0.987
80-90	1.054
90-100	1.085
100-110	0.999
110-120	0.9627
120-130	0.987
130-140	1.054

Table 5.1

Due to the disturbance the terminal voltage was subjected to a transient with small amplitude. Until the 36 th second the controller managed to track the set point satisfactorily. At the 36 th second, the machine is connected to the induction motor load, experiencing a transient with relatively high amplitude. The starting current of the induction motor load was interpreted as a heavy apparent power flow from the machine, so that the voltage transient was significantly high. However adaptive controller is capable of tuning automatically to the correct operating point within a very short time.

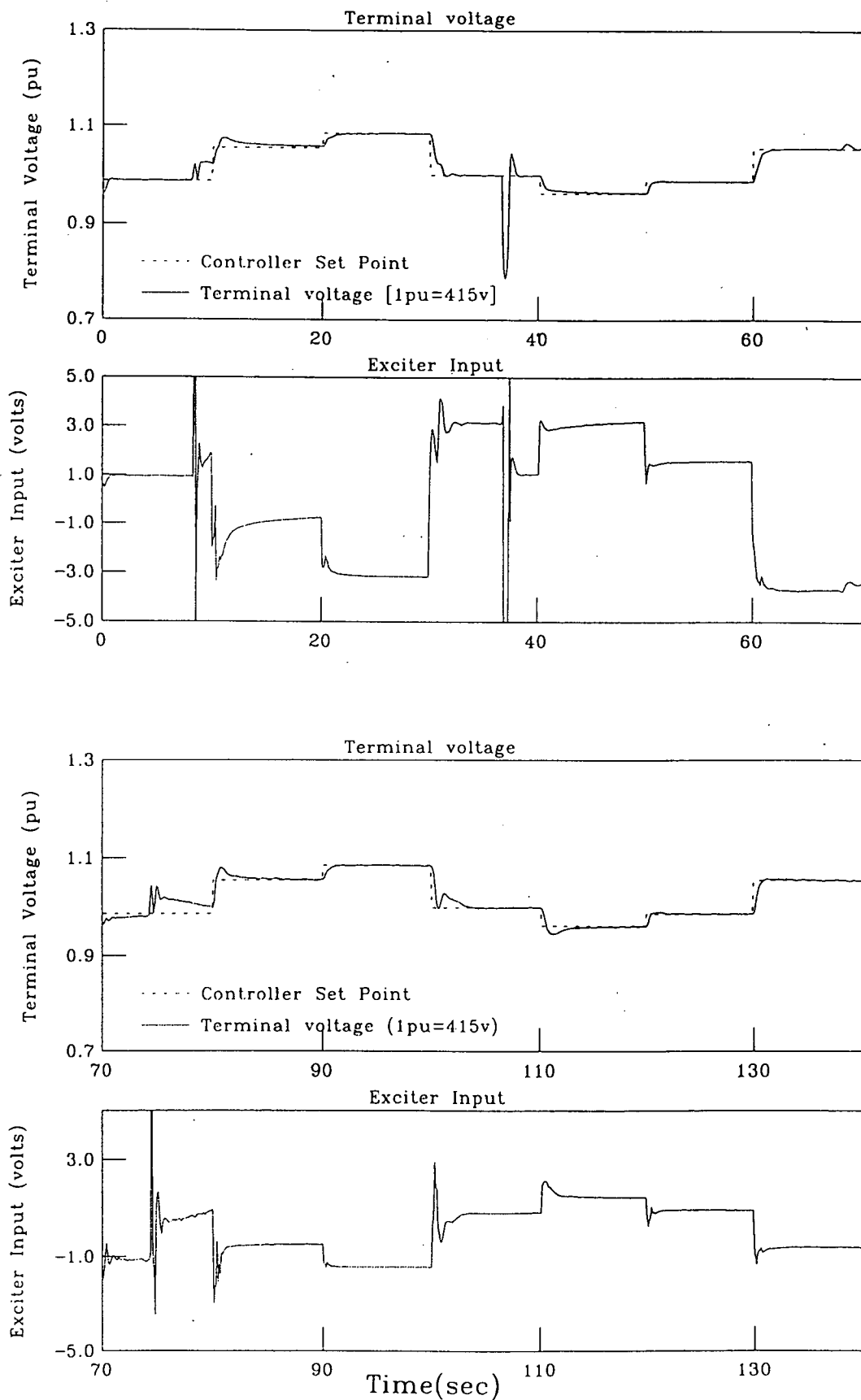


Fig 5.2 Laboratory based adaptive controller - characteristics

At the 74 th second, the machine was isolated form the bus, and a voltage transient of a small amplitude was resulted. Due to the slower response of the governor simulator, it consumed a much longer time interval to return to the perfectly tuned conditions. At the 100 th second the inductive load was removed from the machine. From Fig 5.2.b it can be seen that the machine safely retuned to its new operating conditions.

5.3 Non adaptive (Fixed parameter) exciter controller.

At the second phase of the experiment, a non adaptive controller was implemented using randomly selected plant parameters from the first phase. The plant parameters are,

$$b_1 = -0.00832115$$

$$b_2 = -0.024447$$

$$a_1 = -1.277969$$

$$a_2 = 0.4035884$$

The controller derived from this plant parameters is,

$$u1_{k_1} = -0.32824 u_{sp} - 1.821669 y2_{k_1} + 2.230243 y2_{k_1-1} + 0.15718 u1_{k_1-1} \quad (5.1)$$

$u1$ = exciter input signal (controller output) sample value

$y2_{k_1}$ = terminal voltage sample value

u_{sp} = exciter set point

The computer program FREAL.PAS uses the above equation (5.1) to implement the non adaptive exciter controller. The voltage set points were varied similar to the previous experiment. The response obtained by monitoring the controller for the period of 130 second was plotted in Fig 5.3.

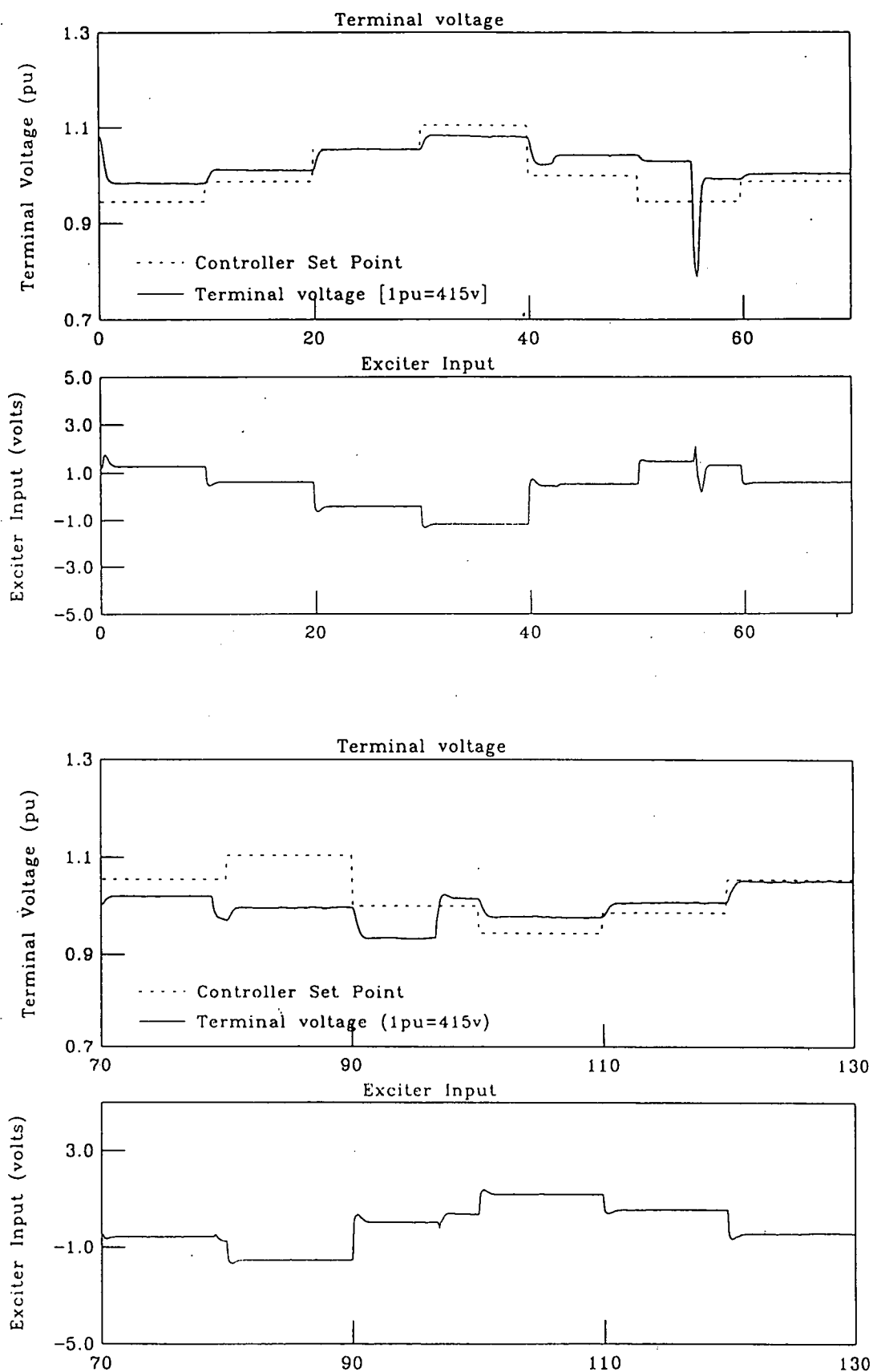


Fig 5.3 Laboratory based fixed parameter controller characteristics.

What follows is an examination of the characteristics in Fig 5.3. The parameters of the plant were obtained with respect to a certain operating point. From Fig 5.3 it can be seen that, the plant perfectly matches with the parameter estimation only the periods between the 20 th second to 30 th second and the 120 th second to 130 th second. At all other times, the controller was unable to track the set point perfectly. The first transient applied to the system was switching to the infinity bus(synchronisation) at the 42 nd second. The resulting effect on terminal voltage due to the transient was comparatively small. At the 55 th second the induction motor load was switched on to the system. The change in terminal voltage due to the transient is relatively high, but returned to a stable value after about one second. At the 78 th second, the infinity bus was isolated from the system and at the 98 th seconds the induction motor load was switched off. It can be seen that the effect of transients does not cause much trouble to the system, even though the controller exhibits incorrect tuning positions at disturbances.

The comparison of the two diagrams 5.2 and 5.3 is straightforward. The adaptive controller is capable of tracking set points irrespective of the operating conditions. Fixed parameter controller tracks set points only at certain system conditions. These experimental results have proved the importance of the adaptive controller.

CHAPTER 6

CONCLUSION

Simulations and the laboratory tests are performed to investigate the feasibility of improving the performance of the exciter control system of a generator by applying adaptive control. Attempts were made to try out the systems with fixed parameter controllers in contrast to the adaptive alternative. A fixed parameter controller is a form of PID controller implemented with a different style.

In most of the cases in the experiment, the plant behaves as a non minimum phase system where open loop zeros lie outside the unit circle. Hence the plant creates an overshoot, but shortly returns to stable conditions. Since the close loop plant is characterised as a non-linear stochastic system, it is very difficult to give general conditions that guarantee the estimates converge. But the simulations in various cases indicate that the algorithm has excellent convergence properties. High feedback gain at low frequencies is a necessity to get a system that is insensitive to low frequency modelling errors and disturbances. This can be achieved by applying an integrator in the control loop. Better results can be expected with signal conditioning by using an anti-aliasing filter before sampling the signals. It limits the unreasonable estimates, due to the dynamics of the system which are unmodelled.

The voltage variations in some of the forced disturbance points seem to be outside the acceptable tolerance. It is basically due to the limitations in the real time system. In commercial power generators, the excitor voltage can rise up to three times higher than the normal operating value to keep the terminal voltage fluctuations within tolerable values. However in a large scale plant, the load is connected to the infinity bus, so that the machine doesn't have to face frequent

load and voltage variations with relatively high magnitudes. In the experiment, the load variations are directly fed back to the governor simulator to significantly highlight system changes. Because the governor simulator is comparatively slower than the exciter, the response of the real time system exhibits longer settling times when the power set point is changed.

It is observed that the self tuning automatic voltage controller based on *pole zero* placement strategy, performs satisfactorily at various operating conditions of the system. Implementation of real time adaptive governor controller is one of the future works that can be proposed as a continuation of this investigation.

Detailed investigations have shown that adaptive control can be useful and give good close loop performance. That doesn't mean that the adaptive control is a universal tool that should always be used. However control engineer should always be aware of those tools to select the best out of many methods.

REFERENCES

1. K. J. Astrom and B. Wittenmark . " Adaptive control" 1989 . Addison Wesley publishing company, USA.
2. P. A.W. Walker and P. H Abdulla "Discrete control of an a.c. turbogenerator by output feedback" , Proceedings I.E.E 1978 Oct Vol.125 pp1031-1038.
3. K. J. Astrom and B. Wittenmark . "Self tuning controllers based on pole zero placement" , Proceedings I.E.E Vol.127 pp120-130.
4. K. J. Astrom and B. Wittenmark . "On Self Tuning Regulators" Automatica Vol. 9, pp. 185-199 Pergamon Press 1973
5. M. J. Grimble ,P. T. K. Fung and M. A. Jonson "Optimum self tuning Control systems: Theory and application" Transactions, institute of measurements and control Vol.4 No1 pp. 25-35 Jan-Mar 1982.

APPENDIX 1

SIMULATED MODEL OF TURBOALTERNATOR.

Synchronous machine is modelled using a set of non linear equations^[2] which is derived by considering the system parameters of a 37.5MVA generator.

A (6x1) state vector x , a (2x1) input control vector U and a (4x1) output measurement vector Y are defined as,

$$X = [\delta, \dot{\delta}, \psi_{fd}, E_{fd}, P_s, T_m]^T \quad (A.1).$$

$$U = [U_e, U_g] \quad (A.2)$$

$$Y = [P_t, v_t, \dot{\delta}, E_{fd}]^T \quad (A.3)$$

Where , δ = Rotor angle in Radians.

ψ_{fd} = Field flux linkage

P_s = Steam power

P_t = Real power outputs at generator terminals

T_m = Mechanical torque input to rotor

u_e = Input to exciter.

u_g = Input to governor

v_t = Terminal voltage

E_{fd} = Field voltage

v_d, v_q = Stator voltages at d and q axis circuits.

i_d, i_q = Stator currents at d and q axis circuits.

ψ_d, ψ_q = Stator Flux linkages at d and q axis circuits.

x_d, x_q = Stator Flux linkages at d and q axis circuits.

x_{ad} = Stator rotor mutual reactance

i_{fd} = Field current

r_{fd} = Field resistant.

e	=	Bus bar voltage
ω	=	Angular frequency of the rotor
ω_0	=	Angular frequency of the infinite bus.
K_d	=	Mechanical damping torque coefficient
τ_e	=	Exciter time constant
τ_g	=	Governor valve time constant
τ_b	=	Turbine time constant
K_v	=	Valve time constant
H	=	Inertia constant
G_v	=	Governor valve position
X	=	State vector (6 x1)
U	=	Input control vector (2 x 1)
Y	=	Output measurement vector (4 x1)

The output vector is selected such that it can be readily measurable in a real Plant.

The non linear system equations of the generator are based on the Park's equations, as defined for a 37.5MVA turboalternator. In addition to the original assumptions made in deriving Park's equations, the following assumptions has also been made.

1. The effect of change of speed and the rate of change of flux linkage in the stator expressions are negligible.
2. Transient effects in the transmission lines are negligible.
3. Negligible magnetic saturation.
4. Line and stator resistances are negligible.
5. Effect of damper winding is counted by adjusting the damping coefficient T_d in the equation of motion.

Considering the system in Fig 3.2, the following equation can be derived as described in Reference 2 .

$$v_d = -\Psi_q \quad (A1.4)$$

$$v_q = \Psi_d \quad (A1.5)$$

$$v_{fd} = R_{fd} \cdot i_{fd} + \dot{\Psi}_{fd} \cdot 1 / \omega_0 \quad (A1.6)$$

$$\Psi_d = x_{ad} \cdot i_{fd} - x_{ad} i_d \quad (A1.7)$$

$$\Psi_q = -x_q \cdot i_q \quad (A1.8)$$

$$\Psi_{fd} = x_{fd} \cdot i_{fd} - x_{ad} i_d \quad (A1.9)$$

$$T_e = \Psi_d \cdot i_q - \Psi_q \cdot i_d \quad (A1.10)$$

$$\ddot{\delta} = \omega_0 / 2H (T_m - T_e - K_d \cdot \dot{\delta}) \quad (A1.11)$$

$$v_t^2 = v_d^2 + v_q^2 \quad (A1.12)$$

Transmission system ,

$$v_d = e \cdot \sin(\delta) - x_e \cdot i_q$$

$$v_q = e \cdot \cos(\delta) + x_e \cdot i_d$$

where x_e = sum of transformer and line reactances.

Prime mover ,

$$\dot{G}_v = U_g / \tau_g - G_v / \tau_g \quad -5 \leq G_v \leq 5$$

$$P_s = K_v \cdot G_v \quad 0 \leq G_v \leq 1$$

$$T_m = (P_s - T_m) / \tau_b \quad (A1.13)$$

Excitation system ,

$$\dot{E}_{fd} = (U_e - E_{fd}) / \tau_e \quad -5 \leq E_{fd} \leq 5 \quad (A1.14)$$

Parameters in a.c. turbo generator ,

$$\text{MVA} = 37.5$$

$$\text{PF} = 0.8 \text{ lagging}$$

$$\text{KV} = 11.8$$

$$\text{RPM} = 3000$$

$$x_d = 2.0 \text{ p.u.}$$

$$x_q = 1.86 \text{ p.u.}$$

$$x_{ad} = 2.0 \text{ p.u.}$$

$$R_{fd} = 0.00107 \text{ p.u.}$$

$$H = 5.3 \text{ MWs / MVA}$$

$$T_d = 0.05$$

$$x_e = 0.470 \text{ p.u.}$$

$$e = 1 \text{ p.u.}$$

$$\tau_e = 0.1 \text{ s.}$$

$$\tau_g = 0.1 \text{ s}$$

$$\tau_b = 0.5 \text{ s}$$

$$K_v = 1.42$$

Constants in a.c Turbogenerator

Defining ,

$$x_d^1 = x_d - x_{ad}^2 / x_{fd} = 0.29;$$

$$x_{dl}^1 = x_d^1 + x_e = 0.79;$$

$$x_{dl} = x_d + x_e = 1.6$$

$$x_{ql} = x_q + x_e = 1.2;$$

$$K_1 = e \cdot x_{ad} / x_{fd} \cdot x_{dl}^1 = 1.256;$$

$$K_2 = e^2 \cdot (x_d^1 - x_q) / x_{dl}^1 \cdot x_{ql} = -0.922;$$

$$K_3 = -r_f \cdot x_{dl} \cdot \omega_0 / x_{fd} \cdot x_{dl}^1 = -0.561;$$

$$K_4 = x_{ad} \cdot \omega_0 \cdot r_f \cdot e / x_{fd} \cdot x_{dl}^1 = 0.422;$$

$$K_5 = x_q \cdot e / x_{ql} = 0.798;$$

$$K_6 = x_e \cdot x_{ad} / x_d^1 \cdot x_{fd} = 0.59;$$

$$K_7 = x_d^1 \cdot e / x_{dl}^1 = 0.365;$$

The non linear relations of the Plant parameters can be justified as,

$$\dot{X}_1 = X_2 \quad (A1.15)$$

$$\dot{X}_2 = [(X_6 - K_1 * X_3 * \sin(X_1) - K_2 * \sin(X_1) * \cos(X_1) - (K_d + T_d) * X_2] * \omega_0 / 2 * H \quad (A1.16)$$

$$\dot{X}_3 = \omega_0 * X_4 * (r_{fd} / x_{ad}) + K_3 * X_3 + K_4 * \cos(X_1) \quad (A1.17)$$

$$\dot{X}_4 = (-X_4 + U_1) / \tau_e \quad (A1.18)$$

$$\dot{X}_5 = (-X_5 + K_v) / \tau_g \quad (A1.19)$$

$$\dot{X}_6 = (-X_6 + X_5) / \tau_b \quad (A1.20)$$

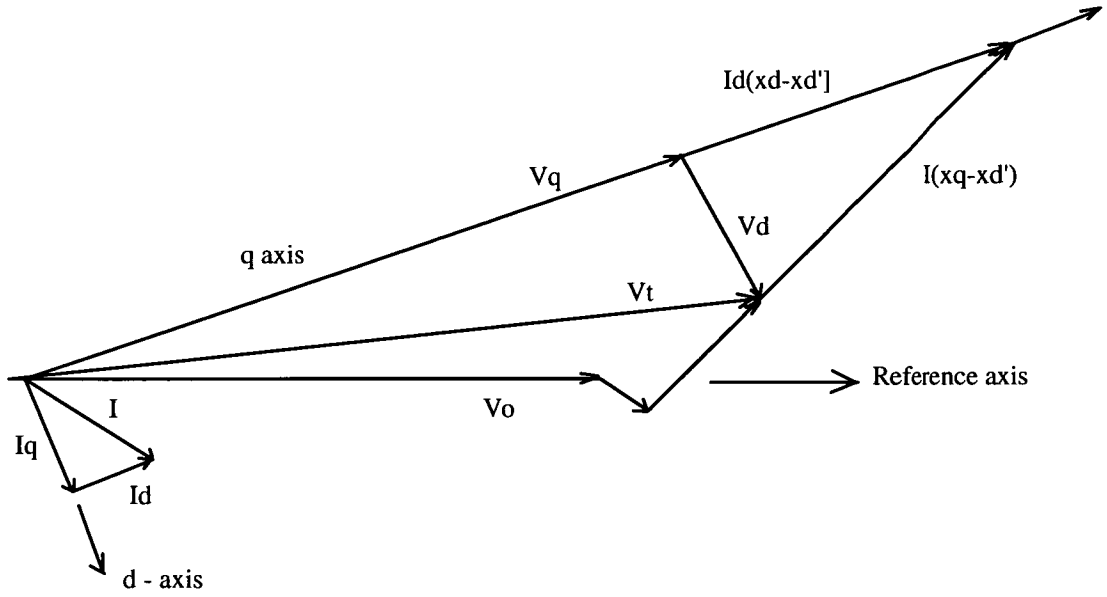


Fig A-2 Vector representation of generator variables.

The outputs Y_1 and Y_2 may be expressed in terms of state variables by ,

$$Y_1 = K_1 * X_3 * \sin(X_1) + K_2 * \sin(X_1) * \cos(X_1) \quad (A1.21)$$

$$Y_2 = (v_d^2 + v_q^2)^{1/2} \quad (A1.22)$$

where,

$$v_d = K_5 * \sin(X_1)$$

$$v_q = K_6 * X_3 + K_7 * \cos(X_1)$$

Above set of equations define the non linear Plant dynamically, so that the solutions of them may describe the Plant state . Equations (A1.15) to (A.20) is solved by the fourth order *Runge Kutta* Integration procedure described in Appendix 3, selecting proper integration step (0.005 in this case). The Synchronous machine can be completely described by the solutions.

APPENDIX 2

RECURSIVE LEAST SQUARES METHOD.

The solutions of a matrix equation $\underline{Y} = \underline{H}\underline{\theta} + \underline{e}$ can be estimated using the least squares' method by calculating the minimum value of the loss function J.

The least square estimate is the value of θ which minimises the square error function(J).

$$J = \frac{1}{2}[e_3^2 + e_4^2 + \dots + e_K^2] \quad (A2.1)$$

$$V = Y - H\theta$$

$$J = \frac{1}{2} V^T \cdot V = \frac{1}{2} (Y - H\theta)^T (Y - H\theta) \quad (A2.2)$$

The value of θ when $\frac{\partial J}{\partial X} = 0$, is denoted as $\hat{\theta}$.

Differentiating with respect to θ ,

$$\frac{\partial J}{\partial \theta} = \frac{1}{2} (H^T (Y - H\theta) + (Y - H\theta)^T \cdot (-H)) = 0$$

$$H^T (Y - H\hat{\theta}) = 0$$

$$H^T Y = H^T H \hat{\theta}$$

$$\hat{\theta} = (H^T H)^{-1} H^T Y \quad (A2.3)$$

Equation A2.3 is called as the normal equation.

Let a vector, P_k as $(H_k^T H_k)^{-1}$

Then, $\hat{\theta}_k = P_k H_k^T Y_k$.

Let the new measurements as Y_{k+1} , and $H_{k+1} = (X_k \ X_{k-1} \ Y_k \ Y_{k-1})$

$$\hat{\theta}_{k+1} = P_{k+1} H_{k+1}^T Y_{k+1}.$$

But,

$$P_{k+1}^{-1} = H_{k+1}^T H_{k+1} = H_k^T H_k + H^T H$$

$$P_{k+1} = (H_{k+1}^T H_{k+1})^{-1} = (H_k^T H_k + H^T H)^{-1}$$

$$P_{k+1} = (P_k^{-1} + H^T H)^{-1}$$

Using the house holder identity (Leema 2.1) ,with $A = P_k^{-1}$, $B = H^T$, $C = H$, the following relations can be obtained .

It can be shown that $\hat{\theta}_{k+1}$ and P_{k+1} can be written in terms of $\hat{\theta}_k$ and P_k as follows.

$$\hat{\theta}_{k+1} = \hat{\theta}_k + K_{k+1} (Y_{k+1} - H_{k+1} \hat{\theta}_k) \quad (A2.4)$$

And,

$$P_{k+1} = P_k - K_{k+1} H_{k+1} P_k \quad (A2.5)$$

Where,

$$K_{k+1} = P_k H_{k+1}^T (1 + H_{k+1} P_k H_{k+1}^T)^{-1} \quad (A2.6)$$

LEEMA 2.1.

HOUSE HOLDER IDENTITY.

This is an algorithm of inverting non singular matrices

Let A, B, C, D non singular square matrices.

Then,

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1}$$

Proof:

$$\text{Define } D = A + BC \quad (1)$$

Premultiplying (1) by D^{-1} ,

$$I = D^{-1}A + D^{-1}BC$$

Post multiplying by A^{-1} ,

$$A^{-1} = D^{-1} + D^{-1}BCA^{-1} \quad (2)$$

$$A^{-1}B = D^{-1}B + D^{-1}BCA^{-1}B = D^{-1}B(I + CA^{-1}B)$$

$$A^{-1}B(I + CA^{-1}B) = D^{-1}B$$

$$A^{-1}B(I + CA^{-1}B)CA^{-1} = D^{-1}BCA^{-1} \quad (3)$$

From (2) and (3),

$$A^{-1} - D^{-1} = A^{-1}B(I + CA^{-1}B)CA^{-1}$$

$$D^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)CA^{-1}$$

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)CA^{-1} \quad (4)$$

APPENDIX 3

RUNGE KUTTA INTEGRATION FORMULA.

Fourth order Runge kutta method can be used to resolve a set of first order non linear equations with a simplified approximation.

Consider the vector equation represented by following expression,

$$\dot{\underline{x}} = A.\underline{x} + B.\underline{u} \quad (A3.1)$$

It is a non linear set in the form of $\dot{\underline{x}} = f(\underline{x}, \underline{u})$

Assume for a very small time period h , $\dot{\underline{x}} = f(\underline{x}, t)$.

Then the estimated value of the vector \underline{x} after time interval h is given by ,

$$\underline{x}(k+h) = \underline{x}(k) + 1/6(K_{-0} + 2.K_{-1} + 2.K_{-2} + K_{-3}) \quad (A3.2)$$

Where,

$$K_{-0} = h.f(\underline{x}, k)$$

$$K_{-1} = h.f(\underline{x} + \frac{1}{2}K_{-0}, k + \frac{1}{2}.h)$$

$$K_{-2} = h.f(\underline{x} + \frac{1}{2}K_{-1}, k + \frac{1}{2}.h)$$

$$K_{-3} = h.f(\underline{x} + K_{-2}, k+.h)$$

The above formulae is called fourth order *Runge Kutta* solution for a non linear set of equations. k is the sample no. which relates to the time.

APPENDIX 4

4.1 PROGRAM LISTING Open loop step response of the synchronous machine :

```
Program OPEN_LOOP_SIMULATION;
{Written by G.P Dissanayaka on 12/12/92}
{#####simulation of synchronous
  machine with Runge Kutta integration
  to identify the Plant dynamics #####}

uses CRT ,GRAPH;
const pi =3.141592;
      h =0.005; {step size for the integration}
type
  vector =array[1..6] of real;

var

  s0,s1,s2,s3      :vector;
  s,x ,xx          :vector;
  i,j,k,l          :integer;
  u1,u2,vd,vq      :real;
  y1,y2            :array[1..610] of real;
  file1            :text;

{   *****   Initial conditions   *****   }

Procedure int_conditions;
Begin
  xx[1] := 1; {delta - rotor angle in radians}
  xx[2] := 0; { derivative of delta}
  xx[3] := 1.152; {flux linkage}
  xx[4] := 2.314; {excitation}
  xx[5] := 0.8; {turbine power}
  xx[6] := 0.8;{mechanical rotor torque PU}
  u1 := 2.495; {exciter set point}
  u2 := 0.634; {governor set point}
End;
```

```
Procedure FUNC;
Begin
```

```

s[1] := x[2];
s[2] := 29.638*(x[6]-1.256*x[3]*sin(x[1])
+0.922*sin(x[1])*cos(x[1])-0.08*x[2]);
s[3] := 0.181*x[4]-0.561*x[3]+0.422*cos(x[1]);
s[4] := 10.0*(-x[4]+u1);
s[5] := 10.0*(-x[5]+1.421*u2);
s[6] := 2.0*(-x[6]+x[5]);

```

End;

Procedure CAL_OF_S;

Begin

```

y1[k] := 1.256*xx[3]*sin(xx[1])
-0.922*sin(xx[1])*cos(xx[1]);{terminal power}

vd := 0.798*sin(xx[1]);{direct axix voltage}
vq := 0.59*xx[3]+0.365*cos(xx[1]);{quadrature axis volltage}

y2[k] := sqrt(vd*vd+vq*vq);{terminal voltage}

for i := 1 to 6 do
  x[i] := xx[i];
  FUNC;
for i := 1 to 6 do
  s0[i] := h*s[i];
for i := 1 to 6 do
  x[i] := xx[i]+0.5*s0[i];
  FUNC;
for i := 1 to 6 do
  s1[i] := h* s[i];
for i := 1 to 6 do
  x[i] := xx[i]+0.5*s1[i];
  FUNC;
for i := 1 to 6 do
  s2[i] := h*s[i];
for i := 1 to 6 do
  x[i] := xx[i]+s2[i];
  FUNC;
for i := 1 to 6 do
  s3[i] := h*s[i];

```

```

writeln(file1,k:2,' ',y1[k]:2:5,' ',y2[k]:2:5,' ',xx[1]:2:5,' ',
xx[5]:2:5,' ');
writeln(k:2,' ',y1[k]:2:5,' ',y2[k]:2:5,' ',xx[1]:2:5,' ',
xx[5]:2:5,' ');
For i := 1 to 6 do Begin
  pp[i] := (1/6)*(s0[i]+2*s1[i]+2*s2[i]+s3[i]);

```

```

xx[i] := xx[i]+(1/6)*(s0[i]+2*s1[i]+2*s2[i]+s3[i]);
End;

```

```

End;

```

```

    {main program}
Begin
int_conditions;

k := 1;assign(file1,'a:openloop.dat');rewrite(file1);
repeat
CAL_OF_S;

k := k+1;
until k>600;readln;

End.

```

4.2 PROGRAM LISTING -Plotting openloop step response:

Program PLOTTING_OPENLOOP_STEP_RESPONSE;

```

{ Written by G.P. Dissanayaka on 7/1/93
  Department of Electrical and Electronic Engineering
  University of Tasmania at Hobart, Tasmania, Australia. }

```

Uses

```

crt,dos,hgrglb, hgrlow, hgrlin, hgraxi, hgrlgn, hgrstr;
{ Graphics routines from HGRAPH }

```

```

const device =0;

```

```

var

```

```

xxi,y1,y2,x1,x5:array[1..610] of real;
file1:text;
count:integer;

```

```

Procedure data_read;

```

```

Begin

```

```

assign(file1,'a:openloop.dat');reset(file1);count := 1;
repeat
readln(file1,xxi[count],y1[count],y2[count],

```



```

        x1[count],x5[count]);
        xxi[count] := xxi[count]/200;
        x1[count] := (x1[count]-1)*180/3.14;
        count := count+1;
    until count=600;
End;
Procedure power_plot; {Plotting results using H-graph software}
Begin

    INIPLT(DEVICE, normal, 1);
    GRAPHBOUNDARY(1500, 8000, 3800, 6800);
    SCALE(0, 3,0.8,0.95);
    SETFONT(BOLD,FALSE);
    JUSTIFYSTRING(5000,7000,'Terminal power',0,2,CENTER,ABOVE);
    AXIS(1, '10.0', 'Time(sec) ', 2, 0.1, '10.1',
        'Terminal power (pu)', 2);
    POLYLINE(xxi, y1, 600, 4, 0, 0, 0, 0);
    ENDPLT;
End;

Procedure voltage_plot; {Plotting results using H-graph software}
Begin

    INIPLT(DEVICE, normal, 1);
    GRAPHBOUNDARY(1500, 8000, 3800, 6800);
    SCALE(0,3,1.105,1.114);
    SETFONT(BOLD,FALSE);
    JUSTIFYSTRING(5000,7000,'Terminal voltage
',0,2,CENTER,ABOVE);
    AXIS(1, '10.0', 'time (sec) ', 2, 0.002, '10.1',
        'Terminal voltage pu', 2);
    POLYLINE(xxi, y2, 600, 4, 0, 0, 0, 0);
    ENDPLT;
End;

Procedure Rot_ang_plot; {Plotting results using H-graph software}
Begin

    INIPLT(DEVICE, normal, 1);
    GRAPHBOUNDARY(1500, 8000, 3800, 6800);
    SCALE(0,3,0,5);SETFONT(BOLD,FALSE);
    JUSTIFYSTRING(5000,7000,'Rotor angle ',0,2,CENTER,ABOVE);
    AXIS(1, '10.0', 'time (sec) ', 2, 1, '10.1',
        'rotor ang (deg)', 2);
    POLYLINE(xxi, x1, 600, 4, 0, 0, 0, 0);
    ENDPLT;
End;

```

Procedure Steam_power_plot; {Plotting results using H-graph software}
Begin

```
INIPLT(DEVICE, normal, 1);  
GRAPHBOUNDARY(1500, 8000, 3800, 6800);  
SCALE(0,3,0.8,0.95);SETFONT(BOLD,FALSE);  
JUSTIFYSTRING(5000,7000,'Steam power ',0,2,CENTER,ABOVE);  
  AXIS(1, '10.0', 'time (sec) ', 2, 0.1, '10.1',  
        'steam power pu', 2);  
  POLYLINE(xxi, x5, 600, 4, 0, 0, 0, 0);  
  ENDPLT;  
End;
```

```
{ ***** Main program ***** }
```

```
  Begin  
    Data_read;  
    Power_plot;  
    Voltage_plot;  
    Rot_ang_plot;  
    Steam_power_plot;  
  End.
```

APPENDIX 5

PROGRAM LISTING - Implimentation of adaptive exciter and governor controller for the simulated turbogenerator

Program Identification_of_Plant_and_adaptive_control;

{ Written by G.P. Dissanayaka on 7/2/93
Department of Electrical and Electronic Engineering
University of Tasmania at Hobart, Tasmania, Australia. }

{ **** Simulation of Recursive least squares method with an exponential
forgetting
factor for the identification of a synchronous machinetransfer function
and
control the output adaptively+** }

{ This program uses the Plant data obtained from the solutions
of the PARK'S EQUATIONS. A set of non linear equations are solved using
4th order
Runge Kutta method selecting integrationtime as 0.005sec. Plant data is used
to predict
the output by applying recursive least squares method with exponential
forgetting factor
as 0.97. The adaptive loop is implimented to control the process }

Uses

crt,dos,hgrglb, hgrlow, hgrlin, hgraxi, hgrlgn, hgrstr;
{ Graphics routines from HGRAPH }

Const delta = 0.97; {exponential fogetting factor}
pi = 3.14159;
h = 0.05; {step size for the integration}
k1 = 3;

Type

matrix = array[1..4,1..4] of real;
vector = array[1..4] of real;
vector1 = array[1..6] of real;

Var

s0,s1,s2,s3 :vector1;
s,x,xx,pp :vector1;
vd,vq,u11,zeeta1,
zeeta2,w1,w2,u22,usp,Psp :real;
q :integer;
p1,p2,kh,ss1,ss2,dummy :matrix;

```

b,hh1,hh2,a,kk1,kk2,px1,px2      :vector;
y2,y1,u1,u2                      :array[1..3] of real;
i,j,l,cc,aa                      :integer;
start                            :boolean;
file1                             :text;

```

```

{ SUBROUTINES FOR MATRIX OPERATIONS }

```

```

Procedure m_unit(var a:matrix; m:integer);
{ creating the identity matrix; dimension = m x m }

```

```

  var i,j:integer;
  Begin
    for i := 1 to m do
      for j := 1 to m do
        Begin
          a[i,j] := 0.0;
          if (i=j) then a[i,j] := 1.0;
        End;
      End;
    End;
  End;

```

```

Procedure v_null(var a:vector; m:integer);
{ creating a vector with zero elements; dimension = m }

```

```

  var i:integer;
  Begin
    for i := 1 to m do
      a[i] := 0.0;
    End;
  End;

```

```

Procedure m_null(var a:matrix; m,n:integer);
{ creating a matrix with zero elements; dimension = m x n }

```

```

  var i,j:integer;
  Begin
    for i := 1 to m do
      for j := 1 to n do
        a[i,j] := 0.0;
      End;
    End;
  End;

```

```

{ ***** End of matrix Procedures ***** }

```

```

Procedure initial_conditions;

```

```

  Begin
    m_unit(p1,4);m_unit(p2,4);
    for i := 1 to 4 do Begin
      p1[i,i] := 100; {**** assumed value for diogonal element of p ****}
      p2[i,i] := 100;
      px1[i] := 0.88;px2[i] := 0.88;End;
    End;
  End;

```

```

xx[1] := 1 ; {delta rotor angle in radians}
xx[2] := 0; { derivative of delta}
xx[3] := 1.152; {flux linkage}
xx[4] := 2.314; {excitation}
xx[5] := 0.8; {turbine power}
xx[6] := 0.8; {mechanical rotor torque PU}
u11 := 2.314; {exciter initial value}
u22 := 0.563; {GOVERNOR SET POINT }
start := true;
usp := 1.1;
psp := 0.80
End;

```

Procedure FUNC1;

Begin

```

s[1] := x[2];
s[2] := 29.637666*(x[6]-1.256*x[3]*sin(x[1])
+0.922*sin(x[1])*cos(x[1])-0.08*x[2]);
s[3] := 0.180726*x[4]-0.561*x[3]+0.422*cos(x[1]);
s[4] := 10.0*(-x[4]+u1[k1]);
s[5] := 10.0*(-x[5]+1.421*u2[k1]);
s[6] := 2.0*(-x[6]+x[5]);

```

End;

Procedure CAL_OF_State;

var nn:integer;

Begin

```

y1[k1] := 1.256*xx[3]*sin(xx[1])
-0.922*sin(xx[1])*cos(xx[1]);
{terminal power }
vd := 0.798*sin(xx[1]); {direct axis voltage}
vq := 0.59*xx[3]+0.365*cos(xx[1]); {quadrature axis voltage}
y2[k1] := sqrt(vd*vd+vq*vq); {calculation of terminal voltage}

```

for i := 1 to 6 do

x[i] := xx[i];

FUNC1;

for i := 1 to 6 do

s0[i] := h*s[i];

for i := 1 to 6 do

x[i] := xx[i]+0.5*s0[i];

FUNC1;

for i := 1 to 6 do

s1[i] := h* s[i];

for i := 1 to 6 do

x[i] := xx[i]+0.5*s1[i];

FUNC1;

for i := 1 to 6 do

s2[i] := h*s[i];

```

for i := 1 to 6 do
  x[i] := xx[i]+s2[i];
  FUNC1;
  for i := 1 to 6 do
    s3[i] := h*s[i];
  for i := 1 to 6 do Begin
    pp[i] := (1/6)*(s0[i]+2*s1[i]+2*s2[i]+s3[i]); {Runge kutta formula}
    xx[i] := xx[i]+pp[i];End;
End;

```

Procedure Plant;

```

{ This Procedure simulates the Plant(synchronous generator) using the
  Fourth order Runge Kutta method using 0.005 sec as the integration
  interval .It uses the results captured from the controller and
  least squares estimation in the previous step. }

```

```

var cyc,nn:integer;

```

```

Begin

```

```

  if start=true then Begin

```

```

    y2[k1-1] := 1.105;cyc := 1;

```

```

    y2[k1-2] := 1.11; {assigned values for starting condition.}

```

```

    u1[k1-1] := u11;

```

```

    u1[k1-2] := u11+0.001;

```

```

    u1[k1] := u11+0.002;

```

```

    u2[k1-1] := u22;

```

```

    u2[k1-2] := u22+0.001;

```

```

    y1[k1] := 0.8;

```

```

    y1[k1-1] := 0.801;

```

```

    y2[k1] := 1.106;

```

```

    start := false;CAL_OF_State;

```

```

  End;

```

```

    if start=false then Begin

```

```

      if cyc=1 then Begin

```

```

        cyc := 0;nn := 9;End

```

```

      else

```

```

        nn := 10;

```

```

      y2[k1-2] := y2[k1-1];

```

```

      y2[k1-1] := y2[k1];

```

```

      u1[k1-2] := u1[k1-1];

```

```

      u1[k1-1] := u1[k1];

```

```

      u2[k1-2] := u2[k1-1];

```

```

      u2[k1-1] := u2[k1];

```

```

      y1[k1-2] := y1[k1-1];

```

```

      y1[k1-1] := y1[k1];

```

```

      for aa := 1 to nn do

```

```

        CAL_OF_State;

```

```

      End;

```

```

End;

```

Procedure H_vector1;

```

Begin
    hh1[1] := u1[k1-1];
    hh1[2] := u1[k1-2];
    hh1[3] := y2[k1-1];
    hh1[4] := y2[k1-2];
End;
{*****start of calculation*****}
Procedure ls_calculation1;
{This Procedure is used for the least squares estimation of parameters.
 calculation is performed for each 0.05 secs.(sample time ).}
var
    t,q,l:double;
    e:double;
Begin
    H_VECTOR1;
    v_null(a,4); { */initialise a */ }
    for i := 1 to 4 do Begin
        b[i] := 0;a[i] := 0;End;
        for j := 1 to 4 do
            for i := 1 to 4 do
                Begin
                    a[j] := hh1[i]*p1[i,j]+a[j];
                    b[j] := hh1[j];
                End;
            q := 0;l := 0;
            for j := 1 to 4 do
                q := a[j]*b[j]+q;
                l := 1/(delta+q); { */application of exponential forgetting factor/* }
                for i := 1 to 4 do
                    Begin
                        t := 0;
                        for j := 1 to 4 do
                            t := p1[i,j]*hh1[j]+t;
                            kk1[i] := t*l;
                        End ;
                    { ***** * / prediction of parameters px /* ***** }
                    e := 0.0;
                    for i := 1 to 4 do
                        e := hh1[i]*px1[i]+e;
                        for i := 1 to 4 do
                            px1[i] := (px1[i]+kk1[i]*(y2[k1]-e));
                    { *****/value of p/***** }
                    m_null(kh,4,4); { ***/initialise kh matrix *****}
                    for i := 1 to 4 do
                        for j := 1 to 4 do
                            kh[i,j] := kk1[i]*hh1[j];

                    {***** square matrix multiplication routine *****}
                    for i := 1 to 4 do

```

```

Begin
  for j := 1 to 4 do
    Begin
      dummy[i,j] := 0;
      for cc := 1 to 4 do
        dummy[i,j] := dummy[i,j]+kh[i,cc]*p1[cc,j];
      End;
    End;
    for i := 1 to 4 do
      for j := 1 to 4 do
        ss1[i,j] := p1[i,j]- dummy[i,j];

      for i := 1 to 4 do
        for j := 1 to 4 do
          p1[i,j] := ss1[i,j]/delta; {***** /assigning new p values *****/}

        End; {End of least square estimation1}

      Procedure H_vector2;
        Begin
          hh2[1] := u2[k1-1];
          hh2[2] := u2[k1-2];
          hh2[3] := y1[k1-1];
          hh2[4] := y1[k1-2];
        End;

        {*****start of calculation*****}
        Procedure ls_calculation2;
        {This Procedure is used for the least squares estimation of parameters .
        calculation is performed for each 0.05 secs.(sample time ).}
        var
          t,qq,l:double;
          e:double;
        Begin
          H_VECTOR2;
          v_null(a,4); { */initialise a */ }
          for i := 1 to 4 do Begin
            b[i] := 0;a[i] := 0;End;
            for j := 1 to 4 do
              for i := 1 to 4 do
                Begin
                  a[j] := hh2[i]*p2[i,j]+a[j];
                  b[j] := hh2[j];
                End;
                qq := 0;l := 0;
                for j := 1 to 4 do
                  q := a[j]*b[j]+qq;
                  l := 1/(delta+qq); {*/application of exponential forgetting factor*/}
                  for i := 1 to 4 do

```



```

Begin
  t := 0;
  for j := 1 to 4 do
    t := p2[i,j]*hh2[j]+t;
    kk2[i] := t*1;
  End ;
{ ***** / prediction of parameters px / ***** }
  e := 0.0;
  for i := 1 to 4 do
    e := hh2[i]*px2[i]+e;
    for i := 1 to 4 do
      px2[i] := (px2[i]+kk2[i]*(y1[k1]-e));
{ *****/value of p/***** }
    m_null(kh,4,4); {***/initialise kh matrix ****}
    for i := 1 to 4 do
      for j := 1 to 4 do
        kh[i,j] := kk2[i]*hh2[j];
{ ***** square matrix multiplication routine ***** }
    for i := 1 to 4 do
      Begin
        for j := 1 to 4 do
          Begin
            dummy[i,j] := 0;
            for cc := 1 to 4 do
              dummy[i,j] := dummy[i,j]+kh[i,cc]*p2[cc,j];
            End;
          End;
        for i := 1 to 4 do
          for j := 1 to 4 do
            ss2[i,j] := p2[i,j]- dummy[i,j];
          for i := 1 to 4 do
            for j := 1 to 4 do
              p2[i,j] := ss2[i,j]/delta; {***** /assigning new p values *****/}
            End; {End of least square estimation2}

```

Procedure Controller1;

```

{-----
This Procedure calculates the controller parameters.
Desired transfer function need following specifications

```

zeeta = Damping coefficient 0.5 - 0.8
hh = Sampling interval.
w = Resonance frequency.

Desired transfer function dominant poles are given by the solution of the equation ,

$$Am = Z^2 - 2 * \cos(w*hh*\sqrt{1 - \text{zeeta}^2})*Z + \exp(-2*\text{zeeta}*w*hh)$$

$$= Z^2 + p_1 Z + p_2$$
Desired T/F is assumed as,

$$G_m(Z) = Z(1+p_1+p_2)/(Z^2 + p_1 Z + p_2) = B_m/A_m$$

```

-----}
var b1,b2,a1,a2,p1,p2,bb,ab,t0,s0,s1,r1 :double;
    hh: double;
Begin
    b1 := px1[1];b2 := px1[2];
    a1 := -px1[3];a2 := -px1[4];
    hh := 10*h; {sampling time=10 times integration interval}
    p1 := -2*exp(-zeeta1*w1*hh)*cos(w1*hh*sqrt(1-sqr(zeeta1)));
    p2 := exp(-2*zeeta1*w1*hh);ab := b2/b1;
    bb := b1;
    if abs(ab)<1 then
        Begin
            {Process zero is cancelled.}

            T0 := (1+p1+p2)/bb;
            S0 := (p1-a1)/bb;
            s1 := (p2-a2)/bb;r1 := 0;
            u1[k1] := T0*Usp - s0*y2[k1]-s1*y2[k1-1]-ab*u1[k1-1];
        End
    else
        Begin
            { Non cancellation of process zero
            Hm(Z) = (1+p1+p2)(Z+ab)/(1+ab)(Z^2 + p1*Z + p2 ) }

            T0 := (1+p1+p2)/(bb*(1+ab));
            R1 := ab-((ab*(ab*ab-p1*ab+p2))/(ab*ab-a1*ab+a2));
            s0 := (p1- a1-r1)/bb;
            s1 := -a2*r1/b2;
            U1[k1] := T0*Usp-s0*y2[k1]-s1*y2[k1-1]-r1*u1[k1-1];
        End;
        if u1[k1]>4.9 then u1[k1] := 4.9;
        if u1[k1]<-4.9 then u1[k1] := -4.9;

    End;
End;

```

Procedure Controller2; {Power controller}

```

var b1,b2,a1,a2,p1,p2,bb,ab,t0,s0,s1,r1 :double;
    hh : double;
Begin
    b1 := px2[1];b2 := px2[2];
    a1 := -px2[3];a2 := -px2[4];
    hh := 10*h; {sampling time=10 times integration interval}
    p1 := -2*exp(-zeeta2*w2*hh)*cos(w2*hh*sqrt(1-sqr(zeeta2)));

```

```

p2 := exp(-2*zeeta2*w2*hh);ab := b2/b1;
bb := b1;
if abs(ab)<1 then
  Begin
    {Process zero is cancelled.}

    T0 := (1+p1+p2)/bb;
    S0 := (p1-a1)/bb;
    s1 := (p2-a2)/bb;r1 := 0;
    u2[k1] := T0*psp - s0*y1[k1]-s1*y1[k1-1]-ab*u2[k1-1];
  End
else
  Begin
    { Non cancellation of process zero
    Hm(Z) = (1+p1+p2)(Z+ab)/(1+ab)(Z*Z + p1*Z +p2 ) }

    T0 := (1+p1+p2)/(bb*(1+ab));
    R1 := ab-((ab*(ab*ab-p1*ab+p2))/(ab*ab-a1*ab+a2));
    s0 := (p1- a1-r1)/bb;
    s1 := -a2*r1/b2;
    U2[k1] := T0*Psp-s0*y1[k1]-s1*y1[k1-1]-r1*u2[k1-1];

  End;
  if u2[k1]>4.9 then u2[k1] := 4.9;
  if u2[k1]<-4.9 then u2[k1] := -4.9;
End;

{***** Main program *****)
Begin

clrscr; assign(file1,'c:\HGRAPH\adv_adp.dat');rewrite(file1);
initial_conditions;textcolor(yellow);
write('INPUT DAMPING COEFFICIENT for exciter control? ');
read(zeeta1);
writeln;
write( 'INPUT RESONANCE FREQUENCY for exciter control (1..4) ? ');
read(w1);
writeln;
write('INPUT DAMPING COEFFICIENT for Governer ? ');
read(zeeta2);
writeln;
write( 'INPUT RESONANCE FREQUENCY for Governer (1...4) ? ');
read(w2);
writeln;

q := 0;

```

repeat

```
if q<600 then usp := 1.2;
if q<500 then usp := 1.15;
if q<400 then usp := 1.1 ;
if q<300 then usp := 1.08;
if q<200 then usp := 1.1 ;
if q<100 then usp := 1.2;
if q<600 then psp := 0.9 ;
if q<450 then psp := 0.85 ;
if q<250 then psp := 0.8;
Plant;
ls_calculation1;
ls_calculation2;
controller1;
controller2;
writeln(u1[k1]:2:3,' ',y2[k1]:2:3,' ',usp:2:3,' ',u2[k1]:2:3,
' ',y1[k1]:2:3,' ',psp:2:3,' ',xx[1]:2:3);

writeln(file1,u1[k1]:2:3,' ',y2[k1]:2:3,' ',usp:2:3,' ',u2[k1]:2:3,
' ',y1[k1]:2:3,' ',psp:2:3,' ',xx[1]:2:3);

q := q+1;
until q=600 ;
```

End.

APPENDIX 6

Program listing - Adaptive exciter control of laboratory based synchronous generator.

{ \$m 32000,0,655360 }

{ \$N+,E+ }

Program ADAPTIVE_CONTROL_project;

Uses

crt,pc30,graph,dos,rtstdhdr,rtgsub,rtgraph,rtscreen;

Const

badd	=\$700; {PC30 PORT ADDRESS}
maxval	= 4096; { Maximum resolution of D/A converters}
delta	= 0.99;{Exponential forgetting factor}
k1	=3;
h	=0.1; {Sampling interval}

Type

Realtype	=real;
matrix	=array[1..4,1..4] of real ;
vector	=array[1..4] of real;

Var

U11, zeeta, w, usp, dummyv	: Realtype;
p,kh,ss,dummy	: Matrix;
b, hh, a, kk, px	: Vector;
u1, y2	: Array[1..3] of realtype;
l, cc, f	: Integer;
start	: Boolean;
yvalues	: Rtvaluearraytype;
tags	: Tagarraytype;
lc, lf	: Rtintarraytype;
miny, maxy	: Realtype;
rt, hialarm, timeint, sampleint	: Realtype;
i, j, nt, grid, count	: Integer;
title	: Titletype;
quit_f	: Boolean;
el, start1, timer, t , m , z	: Integer;
y22, v, ll, cont_output,	
Ter_voltage, setvolt	: Realtype;
file2	: Text;

{ SUBROUTINES FOR MATRIX OPERATIONS required for Plant identification }

Procedure m_unit(var a: matrix; m:integer);

```

{ creating the identity matrix; dimension = m x m }
var i, j: Integer;
Begin
  for i := 1 to m do
    for j := 1 to m do
      Begin
        a[i,j] := 0.0;
        if (i=j) then a[i,j] := 1.0;
      End
    End;
  End;
Procedure v_null(var a:vector; m:integer);
{ creating a vector with zero elements; dimension = m }
var i:integer;
Begin
  for i := 1 to m do
    a[i] := 0.0;
  End;
Procedure m_null(var a:matrix; m,n:integer);
{ creating a matrix with zero elements; dimension = m x n }
var i,j:integer;
Begin
  for i := 1 to m do
    for j := 1 to n do
      a[i,j] := 0.0;
    End;
  End;
Procedure Initial_conditions;
Begin
  m_unit(p,4);
  for i := 1 to 4 do
    p[i,i] := 100; {**** assumed value for diagonal element of p ****}
  End;
  for i := 1 to 4 do
    px[i] := 0.88;u11 := -1.1;
    start := true;
    textcolor(yellow);
    write('INPUT THE VALUE OF THE DAMPING COEFFICIENT ? ');
    read(zeeta);
    writeln;
    write('INPUT THE VALUE OF RESONANCE FREQUENCY (1.4) ? ');
    read(w);
    writeln;
    usp := -1.1;
  End;
Procedure INITQC; {initialising the QUINE CURTIS real time graphics. }
Begin
  for i := 0 to 1 do
    Begin
      lc[i] := 14-i;
      lf[i] := 0;
    End;
  End;

```

```

End;
tags[0] := ";tags[1] := ";
Rtinitgraphics(defaultbgidir,2,1);
Rtsetwintextstyle(rtstat[0],2,4);
Rtsetwintextstyle(rtstat[1],2,4);
Rtinitwindowcolors(rtstat[0],0,7,4,5,17,17,17);
Rtinitwindowcolors(rtstat[1],0,1,4,5,17,17,17);
Rtsetpercentwindow(rtstat[0],0.01,0.01,0.99,0.45);
Rtsetpercentwindow(rtstat[1],0.01,0.51,0.99,0.98);
for i := 0 to 1 do
Begin
hialarm := 15.0;
timeint := 50;
sampleint := 0.1;
grid := 10;
case i of
0:
Begin
rt := 0.99; nt := 1;
title := ' INPUT TO THE EXCITER CONTROLLER';
miny := -10;
maxy := 10;
lc[i] := 13;
End;
1:
Begin
rt := 0.99;nt := 2;
title := ' TERMINAL VOLTAGE';
miny := 350;
maxy := 480;
lc[i] := 2;
End;
End;{End of case}
Case i of
0 : Rtsetupscrollgraph(rtstat[i],timeint,sampleint,miny,maxy,rt,nt,
grid,0.0, hialarm,40.0,1,1, title , 'VOLTS',
tags,lc,lf,false);
1 : Rtsetupscrollgraph(rtstat[i],timeint,sampleint,miny,maxy,rt,nt,
grid,0.0, hialarm,40.0,1,1, title , 'VOLTS',
tags,lc,lf,false);
End;{End of case}
Rtborderwindow(rtstat[i],12);
End;{End of for}
End; {End of Procedure }

```

Procedure ver_chk; {Checking the version of PC 30 hardware}

```

Begin
quit_f := false;
if diag<>0 then

```

```

quit_f := true;
End; {End of Procedure }

```

Procedure DISPLAY(var y22, Ter_voltage:realtype);{data input from the A/D converter}

{ This Procedure is used to monitor the A/D input proportional to the Terminal voltage }

Var

k, ch : Integer;

Begin

ch := 0;

m := ad_in(ch,k);

Ter_voltage := k/maxval;

if ter_voltage>=0.5 then

y22 := (Ter_voltage-0.5)*10

else

y22 := (0.5-Ter_voltage)*(-10);

End; {End of Procedure}

Procedure PLANT;

{ This Procedure processes the Plant data.(synchronous generator)

It uses the results captured from the controller and

least squares estimation in the previous step. }

Begin

if start=true then Begin

y2[k1-1] := 0.5;

y2[k1-2] := 0.51; {assigned values for starting condition.}

u1[k1-1] := u11;

u1[k1-2] := u11+0.001;

u1[k1] := u11+0.002;

y2[k1] := 0.5;

y22 := 0.52;

start := false;

End; {End of if}

if start=false then Begin

y2[k1-2] := y2[k1-1];

y2[k1-1] := y2[k1];

y2[k1] := y22;

u1[k1-2] := u1[k1-1];

u1[k1-1] := u1[k1];

End; {End of if}

End; {End of Procedure}

Procedure H_VECTOR; {Sub_Procedure for ls_calculation}

Begin

hh[1] := u1[k1-1];

hh[2] := u1[k1-2];

hh[3] := y2[k1-1];

hh[4] := y2[k1-2];

End;

Procedure LS_CALCULATION;

{ This Procedure is used for the least squares estimation of parameters.
calculation is performed for each 0.05 secs.(sample time). }

Var

t, qq, l : Realtype;

e : Realtype;

Begin

H_VECTOR;

V_NULL (a,4); { */initialise a */ }

for i := 1 to 4 do Begin

b[i] := 0;a[i] := 0;End;

for j := 1 to 4 do

for i := 1 to 4 do

Begin

a[j] := hh[i] * p[i,j]+a[j];

b[j] := hh[j]; End;

qq := 0; l := 0;

for j := 1 to 4 do

qq := a[j]*b[j] + qq;

l := 1/(delta+qq); { */application of exponential forgetting factor/* }

for i := 1 to 4 do

Begin

t := 0;

for j := 1 to 4 do

t := p[i,j] * hh[j]+t;

kk[i] := t * l;

End ;

e := 0.0;

for i := 1 to 4 do

e := hh[i] * px[i]+e;

for i := 1 to 4 do

px[i] := (px[i] + kk[i]* (y2[k1]-e));

M_NULL (kh, 4, 4); { initialise kh matrix }

for i := 1 to 4 do

for j := 1 to 4 do

kh[i,j] := kk[i ,j] * hh[j];

{ square matrix multiplication routine }

for i := 1 to 4 do

Begin

for j := 1 to 4 do

Begin

dummy [i,j] := 0;

for cc := 1 to 4 do

dummy[i,j] := dummy[i,j]+kh[i,cc]*p[cc,j];

End;

End;

```

    for i := 1 to 4 do
        for j := 1 to 4 do
            ss[i,j] := p[i,j]- dummy[i,j];
    for i := 1 to 4 do
        for j := 1 to 4 do
            p[i,j] := ss[i,j]/delta; { Assigning new p values }
    End; {End of least square estimation}

Procedure Controller; { Calculation of controller output to the machine }
{-----}
    This Procedure calculates the controller parameters.
    Desired transfer function need following specifications
    zeeta = Damping coefficient 0.5 - 0.8
    h = Sampling interval.
    w = Resonance frequency.
    Desired transfer function dominant poles are given by the solution
    of the equation ,
    
$$A_m = Z^2 - 2 * \cos(w * h * \sqrt{1 - \text{sqr}(\text{zeeta})}) * Z + \exp(-2 * \text{zeeta} * w * h)$$

    
$$= Z^2 + p_1 * Z + p_2$$

    Desired T/F is assumed as,
    
$$G_m(Z) = Z(1 + p_1 + p_2) / (Z^2 + p_1 * Z + p_2) = B_m / A_m$$

    -----}
Var b1, b2, a1, a2, p1, p2, bb, ab,
    t0, s0, s1, r1, trace : Realtype;
    err : Integer;
Begin
    count := count+1;
    if count < 100 then usp := -1.0
    else
        if count >= 100 then usp := -0.6;
        if count >= 200 then usp := 0.5;
        if count >= 300 then usp := 1.0;
        if count >= 400 then usp := -0.4 ;
        if count = 500 then count := 1;
        if (count mod 100) = 0 then Begin
            trace := p[1,1]+p[2,2]+p[3,3]+p[4,4];inc(z,1);
            for i := 1 to 4 do
                p[i,i] := 100*p[i,i]/trace; End;
                b1 := px[1];b2 := px[2];
                a1 := -px[3];a2 := -px[4];

            p1 := -2*exp(-zeeta*w*h)*cos(w*h*sqrt(1-sqr(zeeta)));
            p2 := exp(-2*zeeta*w*h);ab := b2/b1;
            bb := b1;
            if abs(ab) < 1 then
                Begin
                    {Process zero is cancelled.}
                    T0 := (1+p1+p2)/bb;
                    S0 := (p1-a1)/bb;

```

```

        s1 := (p2-a2)/bb; r1 := 0;
        u1[k1] := T0*Usp - s0*y2[k1]-s1*y2[k1-1]-ab*u1[k1-1];
    End
else
    Begin
        { Non cancellation of process zero
        Hm(Z) = (1+p1+p2)(Z+ab)/(1+ab)(Z*Z + p1*Z +p2 ) }
        T0 := (1+p1+p2)/(bb*(1+ab));
        R1 := ab-((ab*(ab*ab-p1*ab+p2))/(ab*ab-a1*ab+a2));
        s0 := (p1- a1-r1)/bb;
        s1 := -a2*r1/b2;
        U1[k1] := T0*Usp-s0*y2[k1]-s1*y2[k1-1]-r1*u1[k1-1];
        End;
        if u1[k1] >= 9.5 then u1[k1] := 9.5 ;
        if u1[k1] <=-9.5 then u1[k1] := - 9.5 ;
        End;

Procedure ADAPTIVE_CONTROL;
Begin
    PLANT;
    LS_CALCULATION;
    CONTROLLER;
End;

Procedure OUTPUT(cont_output : Realtype);
{This Procedure outputs controller signals to the D/A channel}
Var
    dac_v, d    : Realtype;
    pp, j, d_ch  : Integer;
Begin
    d_ch := 0;
    d := Cont_output;
    dac_v := maxval/2 + d*maxval/20;
    if dac_v < 0 then dac_v := 1
    else if dac_v > 4095 then dac_v := 4095;  {** check range ** }
    pp := round(dac_v)*-1;
    j := da_out(d_ch,pp);
    End ;
{calculation of limit}

Procedure CALCULATION (var Cont_output : Realtype);
Begin
    if (Cont_output<-9.5) then Cont_output := -9.5;
End;

    { MAIN PROGRAM }
Begin
    Base_30 := badd;
    INITIAL_CONDITIONS;

```

```

VER_CHK ;
Assign(file2, 'c:\tp025\piya\xx1.dat');rewrite(file2);
Assign(file3, 'c:\tp025\piya\xx2.dat'); rewrite(file3);
if not quit_f then Begin
  CNTR_CFG (2);
  AD_CLOCK (20);
  AD_PRESCALAR (10);
  CNTR_WRITE (50000);
  Start1 := 10000;
  Count := 1 ; z := 1 ;
  INITQC; {initialising Q C Graphics}
  While ( Ter_voltage < 2.9) do Begin
    CNTR_WRITE (50000);
    Start1 := CNTR_READ;
    ADAPTIVE_CONTROL ; {which gives the output of the controller
      cont_output}
    CALCULATION (Cont_output);
    OUTPUT (Cont_output);
    DISPLAY (y22,Ter_voltage);
    yvalues[0] := Cont_output;
    Rtupdatescrollgraph (rtstat[0], yvalues);

    dummyv := y22*25.4+424.94;
    setvolt := usp*25.4+424.94;

    if (z >= 6) and ( Z < 12 ) then writeln (file2, Cont_output:2:3,
      '', dummyv:2:3,' ', Setvolt:2:3,' ',
      px[1],' ',px[2],' ',px[3],' ',px[4]);
    if (z >= 14 ) and ( Z < 20 ) then writeln( file2,Cont_output:2:3,
      '',dummyv:2:3,' ',Setvolt:2:3);

    yvalues[0] := dummyv;
    yvalues[1] := Setvolt;
    Rtupdatescrollgraph ( rtstat[1], yvalues);

    if z > 20 then Begin
      Close(file2);
      halt;
      End;
    repeat
      TIMER := CNTR_READ;
      EL := START1-TIMER
      if el<0 then START1 := 50000+START1;
      until el >= SAMPLEINT * 10000;
    End;

    Cont_output := -0.60;

```

```
CALCULATION(Cont_output);  
OUTPUT(Cont_output);
```

```
End;  
Rtclosetgraphics (1);
```

```
End.
```