APPLICATION OF QUANTITATIVE FEEDBACK THEORY TO ROBUST POWER SYSTEM STABILISER DESIGN

Paramasivan Chetty

Submitted in fulfilment of the requirements of Master of Science in Engineering, in the School of Electrical, Electronic, and Computer Engineering, University of Natal, on March 25, 2003

I hereby declare that the material incorporated into this thesis is my own original and unaided work except where specific reference is made by name or in the form of a numbered reference. The work contained herein has not been submitted for a degree at any other university.

Paramasivan Chetty

Dedication

To my Dad, Mum, and Sister

Abstract

This thesis aims to verify the use of quantitative feedback theory (QFT) as a viable tool for designing power system stabilisers (PSS) for a single machine infinite bus system.

The result of the QFT design is verified by simulation of the linear and nonlinear models representing the power system, and also by experimental procedures carried out in a laboratory.

QFT falls into the classical control category, and is a frequency domain design method. It is an alternative to other design methods such as root locus and H_{∞}. The QFT design procedure can be extended to a multi-machine system and QFT designs of MIMO systems has gained impetus.

From theory, through simulation, and to the final laboratory testing on a single machine, infinite bus system, it will be shown that the application of QFT to robust PSS design does indeed work.

QFT is a design method that allows the designer to choose a set of realistic operating points and to produce a design that include those points. Other methods allow the designer to produce a design for single operating point, and one has no idea how the design performs at the other operating points.

Acknowledgements

I wish to express my gratitude to the University of Natal and to the National Research Foundation for their interest and sponsorship.

A special thank you goes to my mentor, Dr. Edward Boje, whose encouragement and wise advice made me gain an appreciation for control theory.

I would like to thank Dr. B.S Rigby, and the staff at the Electrical Workshop in the School of Electrical and Electronic Engineering, for their time and assistance with the power system laboratory.

Finally, I wish to acknowledge my family for their invaluable support during the writing of this thesis.

Contents

List of Figures

List of Tables

Nomenclature

$\dot{x}$	first derivative of x
$\ddot{x}$	second derivative of x
δ	load angle
λ	magnetic flux linkage
$\mathcal{B}$	QFT boundary set
$\mathcal{L}$	loop transmission set
$\mathcal{P}$	plant set
P	number of poles in the synchronous machine.
$\boldsymbol{i_{abc}}$	instantaneous 3 phase current vector
$\boldsymbol{i_{qd}}$	instantaneous *d*-*q* transformation current vector
$\boldsymbol{K_s}$	*d*-*q* transformation matrix
$\boldsymbol{L_r}$	rotor inductance matrix
$\boldsymbol{L_s}$	stator inductance matrix
$\boldsymbol{R_r}$	rotor resistance matrix
$\boldsymbol{R_s}$	stator resistance matrix
$\boldsymbol{v_{abc}}$	instantaneous 3 phase voltage vector
$\boldsymbol{v_{qd}}$	instantaneous *d*-*q* transformation voltage vector
ω	speed in radian per second
ψ	magnetic flux
τ''_{d0}	*d-axis* subtransient open circuit time constant
τ''_{d}	*d-axis* subtransient short circuit time constant

τ''_{q0}	*q-axis* subtransient open circuit time constant
τ''_{q}	*q-axis* subtransient short circuit time constant
τ'_{d0}	*d-axis* transient open circuit time constant
τ'_{d}	*d-axis* transient short circuit time constant
τ_{ϵ}	exciter time constant
τ_{W}	washout time constant
τ_{kd}	*d-axis* damper leakage time constant
θ	angle of d-q reference frame
θ_{ef}	angle of electrical quantity
D	mechanical damping constant
E_{fd}	field voltage
f	electrical quantity
$G(s)$	controller transfer function
H	inertia constant
i_d	instantaneous *d-axis* current
i_q	instantaneous *q-axis* current
i_{fd}	instantaneous *d-axis* field current
i_{kd}	instantaneous *d-axis* damper winding current
i_{kq}	instantaneous *q-axis* damper winding current
J	moment of inertia
K	mechanical viscous constant
K_{ϵ}	exciter gain
K_a	AVR gain
$L(s)$	loop transmission function
L_l	armature leakage inductance
L_n	nominal loop transmission function
L_t	inductance of transmission line

L_{lfd}	leakage inductance of field winding
L_{lkd}	leakage inductance of *d-axis* damper winding
L_{lkq}	leakage inductance of *q-axis* damper winding
L_{md}	*d-axis* magnetising inductance
L_{mq}	*q-axis* magnetising inductance
$P(s)$	plant transfer function
P_a	active power in watts
P_b	active power in per unit
P_e	electrical power
P_n	nominal plant function
q	coulomb charge
Q_a	reactive power in vars
Q_b	reactive power in per unit
R_t	resistance of transmission line
s	Laplace operator
T_e	electrical torque
T_m	mechanical torque
V_b	infinite bus voltage
v_d	instantaneous *d-axis* voltage
v_q	instantaneous *q-axis* voltage
V_t	terminal voltage
v_{fd}	instantaneous *d-axis* field voltage
v_{kd}	instantaneous *d-axis* damper winding voltage
v_{kq}	instantaneous *q-axis* damper winding voltage
V_{ref}	reference voltage
X'_d	*d-axis* transient reactance
X''_d	*d-axis* subtransient reactance

X_q''	*q-axis* subtransient reactance
X_d	*d-axis* synchronous reactance
X_f	field leakage reactance
X_l	armature leakage reactance
X_q	*q-axis* synchronous reactance
X_t	reactance of transmission line
X_{kd}	*d-axis* damper leakage reactance
X_{kq}	*q-axis* damper leakage reactance
X_{md}	*d-axis* magnetising reactance
X_{mq}	*q-axis* magnetising reactance

Chapter 1

Introduction

> Hypotheses non fingo. I feign no hypotheses.
>
> ISAAC NEWTON
> *Principia Mathematica*

Synchronous machines generate most of the world's electrical energy, and as such, studies of their operation and stability play a major rôle in ensuring that we get the maximum power output from existing machinery. Most utilities run their generators at the maximum stability limit during peak demand. Such operation reduces the stability margins inherent in the machines. As the demand for the machines to supply varying loads grow, so does the need to develop controls which are reliable, robust and can be implemented cost-effectively. In addition, such controls must also be able to provide the necessary compensating effect to counteract the reduction in stability margins.

1.1 The PSS Problem

The low frequency local mode mechanical oscillations that are excited by a change in operating conditions can be defined as the power system stability problem. The main focus lies in the stability of the torque-angle loop, i.e. the behaviour of the rotor angle and speed after it has been subjected to a small signal perturbation. A mechanical torque disturbance from the prime mover is one such perturbation.

This phenomenon has been the topic of interest and much research recently [7], [34], [19]. The usual method of counteracting such behaviour has been to damp out the rotor oscillations with a supplementary signal derived from the machine speed. This signal is added to the voltage reference signal, and the resulting signal generates a component of electrical torque (synchronising torque) that is in phase with the rotor speed deviations.

Since the frequency of oscillation is dependent on the synchronising torque, when the PSS loop is closed, one should ensure that the voltage–speed transfer function eigenvalues magnitude do not decrease appreciably. Power systems rarely operate at a fixed operating point. Plant parameter variations due to changes in generation and load patterns, as well as transmission networks means that there are correspondingly (possibly large) variations in the small signal dynamic behaviour. Designing a PSS properly greatly enhances the overall performance of the system, but some of these designs are valid only for the designed operating point. Therefore, tuning a PSS becomes a difficult task due to the constantly changing dynamic behaviour of the system.

There have been attempts in developing artificial neural network, fuzzy logic or adaptive power system stabilisers that deal with these parametric variations, but they are limited in their application due to their complexity and computational requirements. The most popular form of PSS used in industry is the conventional lead compensation power system stabiliser [21]. It offers ease of use and is readily implemented as an analogue circuit [19].

The change in operating conditions can be represented as a parametric variation in the linearised model of the power system. Such uncertainty can be handled by quantitative feedback theory. Therefore, QFT is ideally suited to designing a controller that is robust despite operating condition variations. The controller parameters are determined by solving the required stability and performance specifications. It is imperative that the stability criterion is satisfied at all operating conditions to ensure robustness. Performance specification in the power systems field usually relates to the damping factor required.

Irrespective of the design technique used in developing power system stabilisers, it is necessary to recognise the nonlinear nature of the power system and that the objective of the PSS is to extend the power transfer limits by damping rotor oscillations.

The PSS does not enhance transient stability. In fact, it will often have a detrimental effect on transient stability by attempting to pull the generator field out of ceiling too early in response to a fault [24]. However, PSS tuning does have an impact upon system performance following a small signal disturbance. This is accomplished by modulating the voltage regulator setpoint such that the resulting torque changes are in phase with the shaft speed.

Chapter 2

Literature Survey

> To Thales the primary question was not what we know, but how do we know it.
>
> ARISTOTLE
> *Mathematical Intelligencer*

The aim of this survey is to highlight the research efforts undertaken to identify, analyse and implement power system stabilisers and other control strategies such as excitation and speed control that improves the dynamic performance of the system. The survey traces the historical developments in power system stabiliser design. The survey deals predominantly with the work done in the last eight years. For completeness, the celebrated paper by DeMello and Concordia has also been included.

2.1 1969

DeMello and Concordia [10] treat the stability of a synchronous machine from the viewpoint of linearised small signal perturbations, as shown in Figure 2.1. The constants K_1 to K_6 depend on the machine parameters and operating point. The calculations for these constants can be found in most power system textbooks. From Figure 2.1, the stability limit without a PSS is the condition for which $K_1 - K_2K_4K_3$ is greater than or equal to zero, assuming constant field voltage. Of particular interest is the transfer function from ΔT_m to $\Delta\delta$. This transfer function describes the effect of the load angle on the terminal voltage, and is given by Equation 2.1.

$$\frac{\Delta T_m}{\Delta\delta} = \frac{-K_2K_\epsilon K_5}{\left(\frac{1}{K_3} + K_6K_\epsilon\right) + \left(\frac{\tau_\epsilon}{K_3} + \tau'_{do}\right)s + \left(\tau'_{do}\tau_\epsilon\right)s^2} \tag{2.1}$$

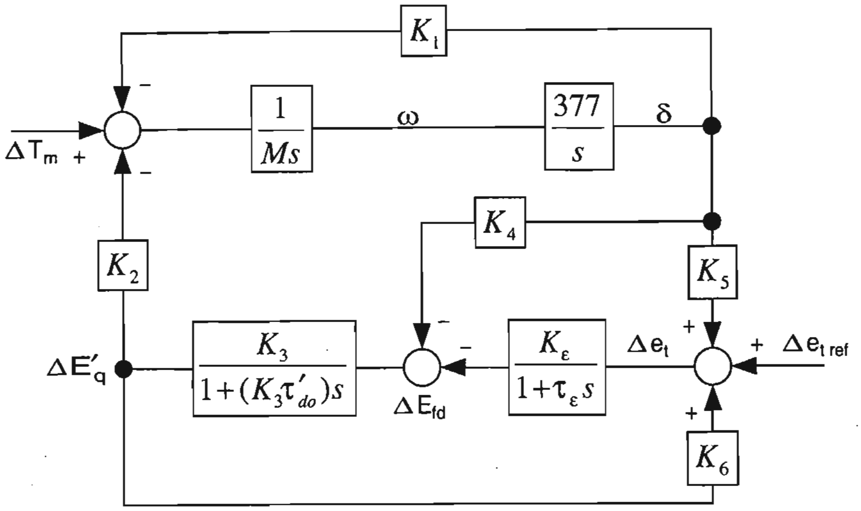


Figure 2.1: Small signal linear model, including the effect of the voltage regulator-excitation system as used for stability studies.

For very low frequencies, Equation 2.1 becomes

$$\frac{\Delta T_s}{\Delta\delta} = \frac{-K_2K_\epsilon K_5}{\left(\frac{1}{K_3}+K_6K_\epsilon\right)} \tag{2.2}$$

This is the synchronising torque, and for high exciter gain values, $K_3 \to \infty$

$$\frac{\Delta T}{\Delta\delta} = \frac{-K_2K_5}{K_6} \tag{2.3}$$

K_5 is positive when there is low to medium external impedance and low to medium loading. In this case, K_1 is usually high so the stability condition is still greater than zero and the system is stable. K_5 is negative when there is moderate to high external impedance and heavy loading. The stability condition is positive, and the system is stable. This is useful when the K_1 for a particular machine is low or negative, or when the stability condition is negative without a regulator.

The damping torque component magnitude due to regulator action is

$$\left|\frac{\Delta T}{\Delta\delta}\right| = \frac{K_2K_\epsilon K_5\left(\frac{\tau_\epsilon}{K_3}+\tau'_{do}\right)\omega}{\left(\frac{1}{K_3}+K_6K_\epsilon-(\tau'_{do}\tau_\epsilon)\omega^2\right)^2+\left(\frac{\tau_\epsilon}{K_3}+\tau'_{do}\right)^2\omega^2} \tag{2.4}$$

With K_5 positive, the component gives positive damping. With K_5 negative, and this is usually the case for most modes of operation, the damping torque is negative. Also, increasing K_ϵ increases the magnitude of the negative damping, causing instability. When K_5 is negative, a voltage regulator is useful in providing synchronising torque, and stabilising the machine. It does this at the expense of lowering the machine's natural damping, which is usually small. The solution has been to set the regulator gain such that it provides the necessary synchronising torque without cancelling all the inherent machine damping. There will be some instances where there is poor damping, as in the case of long lines operating near the line limit. In these cases, providing an additional stabilising signal derived from the machine speed, terminal voltage frequency, or accelerating power can solve the problem.

To quote DeMello and Concordia:

> It is evident that for every combination of machine and system parameters, and loading conditions, there are a variety of stabilising signal transfer functions operating on speed which give essentially the right phase and magnitude relation for the frequency of concern. It is a challenge, however, to find a universal function which would be adequate for the whole spectrum of possibilities.

Design of generator excitation systems was also explored by Hughes and Hamdan [16]. The method of design was by multivariable frequency response techniques, since it was discovered that the speed and voltage loops cannot be treated independently. A little later, some work was carried out to implement a supplementary controller to augment the existing excitation system using an Inverse Nyquist Array [2].

2.2 1980

Larsen and Swann [24] extended the work of DeMello and Concordia in the 1980's. Having identified the single machine infinite bus PSS problem, they developed a method of tuning power system stabilisers in the field. This method is still widely used today [44], [23]. With the inclusion of a PSS the diagram shown in Figure 2.1 now becomes Figure 2.2(b). The electrical torque component now has two distinct components; one which is produced solely by the contribution of the PSS, and the other component which results from the sum of all other torques. The transfer function relating the torque input to the speed output, and due only to the PSS is

$$\frac{\Delta T_{ep}}{\Delta\omega} = PSS_\omega(s)GEP(s) \triangleq P(s) \tag{2.5}$$

where $GEP(s)$ collectively represents the transfer functions of the exciter, the generator, and the transmission system. When the generator speed is

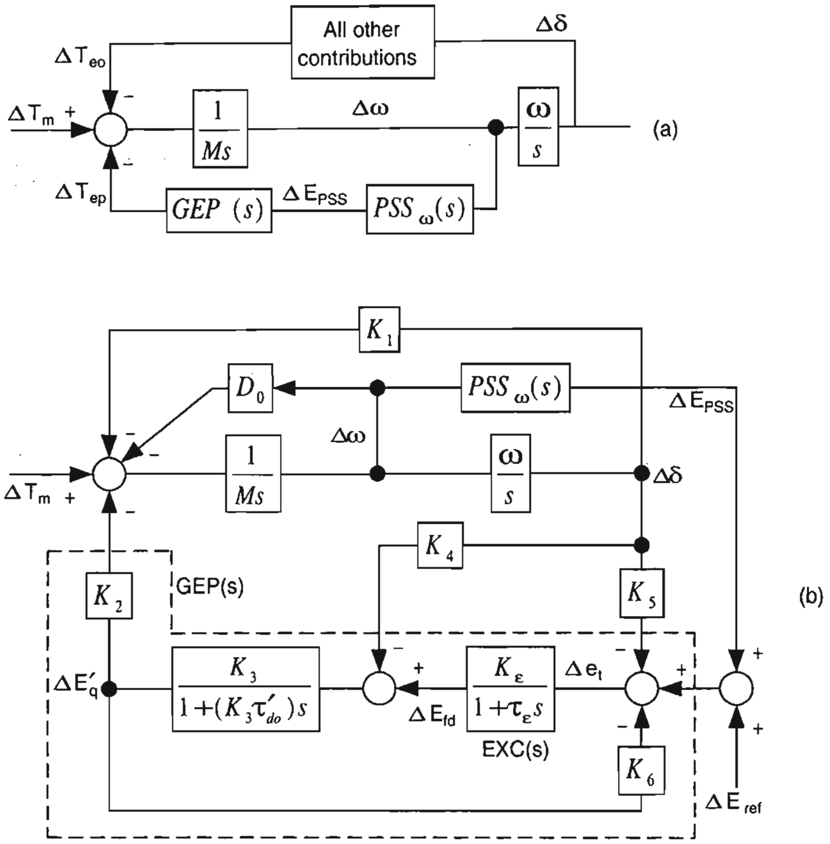


Figure 2.2: (a) Simplified block diagram showing the contribution of the PSS. (b) Detailed model showing the addition of a PSS.

constant, $GEP(s)$ can be simplified to

$$GEP(s) \cong \frac{K_2}{K_6} \frac{\partial E'_q}{\partial E_{ref}} \tag{2.6}$$

This relationship forms the basis of PSS tuning using the closed loop AVR characteristic to determine the phase compensation required. The variation of $GEP(s)$ with respect to exciter gain, line loading and ac network stiffness plays a major role in the PSS design and performance.

For cases where the AVR loop crossover frequency is below the electromechanical oscillation frequency of interest, the gain of $GEP(s)$ can be approximated by

$$|GEP(s)| \cong K_2 \frac{|EXC(s)|}{\tau'_{do} s} \tag{2.7}$$

It can be seen that the gain of $GEP(s)$ is dependent on K_2. K_2 represents the effect of a change in generator flux E'_q on torque and increases proportionally [25] as

1. the ac network stiffness increases, and
2. the generator loading increases.

The gain is therefore highest when the generator is operating into the strongest ac system at full load.

For cases when the AVR loop crossover frequency is above the frequency of interest, Equation 2.6 becomes

$$|GEP(s)| \cong \frac{1}{K_6} \frac{|EXC(s)|}{\tau'_{do} s} \tag{2.8}$$

The gain is now inversely proportional to K_6, where K_6 represent the effect of generator flux E'_q on terminal voltage E_t. K_6 decreases, as the ac network becomes stronger. Thus the gain of $GEP(s)$ increases and the network stiffness increases. However, it can be seen from Equation 2.8 that the crossover frequency decreases, as the network becomes stronger. This influences PSS design, since there is more phase lag to overcome with a strong network than with a weak one.

The input signals to a power system stabiliser can be speed, frequency, or accelerating power. A PSS using shaft speed as input must compensate for the lag in $GEP(s)$ to produce a component of torque in phase with speed deviations. The ideal PSS would therefore be inversely proportional to $GEP(s)$

$$PSS(s)_{ideal} = \frac{D_{PSS}}{GEP(s)} \tag{2.9}$$

where D_{PSS} represents the desired damping transfer function. However such a controller is impractical, since compensation requires pure differentiation with its associated high gain at high frequency. Also, uncertainty cannot be handled with a fixed controller. The practical approach is to utilise lead/lag networks to compensate for phase lags over the frequency range of interest. A washout stage is usually added to eliminate steady state voltage offsets as the system frequency changes, otherwise voltage following is affected. The transfer function now becomes

$$PSS(s)_{practical} = K_5 \frac{\tau_W s}{1 + \tau_W s} \frac{(1 + \tau_1 s)(1 + \tau_3 s)}{(1 + \tau_2 s)(1 + \tau_4 s)} \tag{2.10}$$

AC bus frequency is used as the input in cases where the ac network strength is weak. This arises from the fact that the sensitivity of the frequency signal to rotor oscillation increases as the external transmission system becomes weaker, which tends to offset the reduction in gain from PSS output to electrical torque. In addition, the frequency signal is more responsive to inter-area modes, rather than local modes. This means it is possible to get greater damping between plants with a frequency input signal than with a speed input signal.

When accelerating power is used as an input signal, the PSS can be designed with a non-minimum phase characteristic. The approach is to add appropriate phase lag networks such that the net phase is −360° at the frequency of concern. This results in a "non-minimum phase" control loop [24]. For example, in a system requiring 45° phase lead to achieve stability, a lag term can be used to give 315° phase shift rather than a lead term which gives 45° phase shift. This allows lag networks to be used, rather than the lead networks usually used with speed or frequency input. The effect is similar to a speed input PSS, but with lower gain at high frequencies. The PSS is, however, sensitive to low frequency phenomena, such as changes in mechanical power. The PSS must have compensation for these phenomena. If a washout stage is used as compensation, then phase lead is introduced at low frequency, and may cause instability of inter-area swings.

Field tuning of power system stabilisers and practical considerations can be found in [25] and [26].

2.3 1993

A practical investigation into the best PSS input signal to use was carried out at the Blue Mesa power plant in 1993. The Colorado Bureau of Reclamation commissioned this feasibility study. Hoa and Agee [44] documented their field tests and simulation results and concluded that all three input signals can be used to damp local mode oscillations. However, the accelerating power input signal gave the best system performance for inter-area

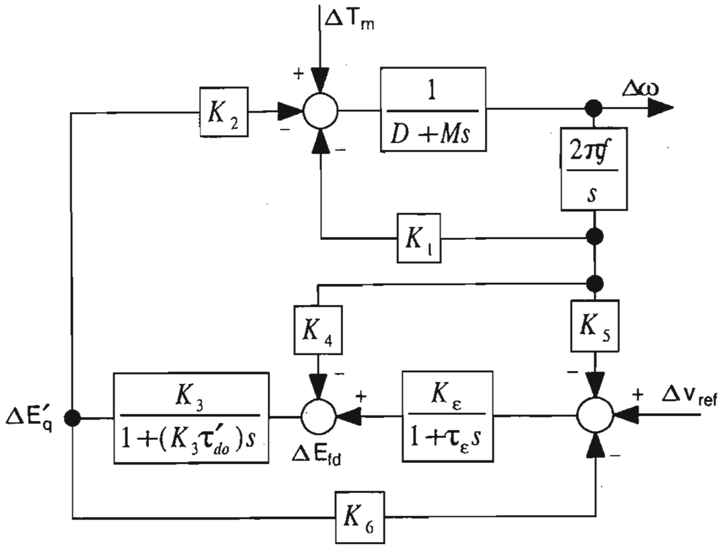


Figure 2.3: The fourth order linearised model used in the QFT design.

oscillations.

The first formal treatment of PSS design using quantitative feedback theory appeared in 1994. Sedigh and Alizadeh [38] used a single machine connected to an infinite bus as their plant model. This fourth order model did not take into account the damper winding and the effect of the governor. The block diagram representation of the linear model is shown in Figure 2.3. This is roughly equivalent to Figure 2.2(b).

The performance specifications called for the voltage reference tracking control ratio to be between upper and lower bounds. These bounds are defined in the paper. The specifications state that for a step input of 0.02 per unit (p.u), the maximum overshoot will be 20e–3 p.u. The plant is assumed to be stable at all operating conditions, since no mention was made of stability bounds. Uncertainty range was assumed on all the model parameters, except the frequency.

The following steps were carried out to arrive at the PSS transfer function:

1. forming the plant templates: the magnitude and phase of the model P_0 at 3 different operating points was calculated for each frequency in the range $[0.1 \ldots 1000]$,

2. forming the $\mathcal{U}$ - contour: this is the stability boundary on the Nichols chart which the open loop transfer function must be outside,

3. forming the performance boundaries,

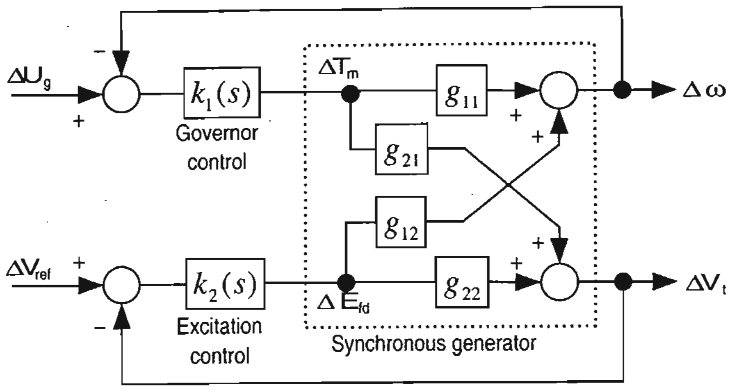


Figure 2.4: Small signal structure of a generator showing the cross-couplings between the voltage and speed loops.

4. shaping the loop transmission function L_0, and
5. calculating the PSS transfer function from

$$G(s) = \frac{L_0}{P_0} \tag{2.11}$$

The QFT PSS was compared to a state feedback PSS. The transient response of the QFT design was stable at all operating conditions, whereas the state feedback controller gave poor transient response.

2.4 1998

Fadlalmoula *et al* [12] provides an analysis of the effect of the addition of a PSS from a control engineering perspective. The use of multivariable techniques allows the system to be treated as s pseudo SISO system where the speed loop is first closed, and the AVR loop is designed for the purpose of rejecting voltage disturbances. The model used to show the cross coupling between the speed and voltage loops, or channels, is shown in Figure 2.4.

The synchronous generator is modelled as the popular 3rd order system, which is accurate enough to highlight the most significant dynamics. The input-output relationship for the system is given as

$$\boldsymbol{G} = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} = \frac{1}{\Delta s} \begin{bmatrix} \bar{g}_{11}(s) & \bar{g}_{12}(s) \\ \bar{g}_{21}(s) & \bar{g}_{22}(s) \end{bmatrix} \tag{2.12}$$

where the elements of the second matrix can be found in explicit form in the paper. After some multivariable analysis, Figure 2.4 can be redrawn as

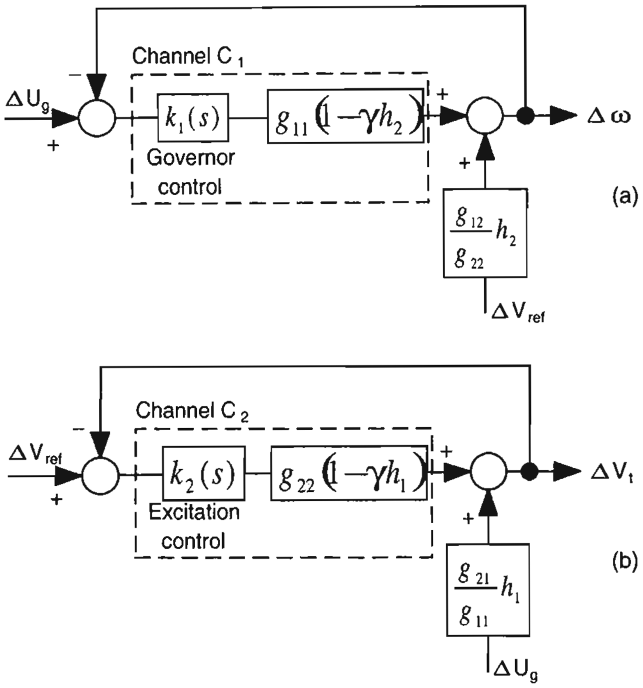


Figure 2.5: Individual multivariable loop analysis of (a) speed-governor control loop and (b) AVR-excitation control loop.

Figure 2.5. The multivariable structure of the SISO channels C_1 and C_2 is described by the complex frequency structure $\gamma(s)$. When $\gamma(s) \ll 1$, the signal interaction between the loops is small, and when $\gamma(s)$ is large, the signal interaction between the loops is high. $\gamma(s)$, h_1 and h_2 are given by

$$\begin{gathered} C_1 = k_1 g_{11}(1 - \gamma h_2), \quad C_2 = k_2 g_{22}(1 - \gamma h_1) \\ \gamma(s) = \frac{g_{12}g_{21}}{g_{11}g_{22}}, \quad h_1(s) = \frac{k_1 g_{11}}{1 + k_1 g_{11}}, \quad h_2(s) = \frac{k_2 g_{22}}{1 + k_2 g_{22}} \end{gathered} \tag{2.13}$$

It was formally confirmed that the slow governor loop has little effect on the exciter loop response, and therefore the performance of the system is mostly determined by the electrical subsystem transfer function. Looking at the frequency response of the electrical subsystem, a "switchback" characteristic due to a lightly damped pole pair followed by a lightly damped zero pair was noted. Applying direct control to the electrical subsystem over a frequency range that includes this "switchback" will result in instability. Also, the frequency at which this "switchback" occurs changes with the loading condition, thus making direct control for different load conditions impossible. However there exists two control options. They are

1. AVR-speed control without a PSS, and
2. AVR-speed control with a PSS.

Using control option 1, the switchback is avoided by using a slower response excitation system. Comparison of the open loop gain crossover frequency response of the AVR-excitation system is not unlike that of the open loop governor subsystem. Considering the system as two SISO loops is valid in this case. The drawback of using this method is the poor disturbance rejection properties of the AVR-excitation system at higher frequencies.

Control option 2 seeks to increase the bandwidth of the AVR-excitation system, to provide better disturbance rejection at the higher frequencies. This new frequency range will include the "switchback" characteristic, which is undesirable. With the PSS in place, however, the new system becomes

$$\begin{aligned} \boldsymbol{G'} &= \boldsymbol{PG} \\ &= \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) + p(s)g_{11}(s) & g_{22}(s) + p(s)g_{12}(s) \end{bmatrix} \\ \text{where} \quad \boldsymbol{P} &= \begin{bmatrix} 1 & 0 \\ p(s) & 1 \end{bmatrix} \\ \text{and} \quad p(s) &= k_s \frac{\tau_W s}{1 + \tau_W s} \frac{1 + \tau_1 s}{1 + \tau_2 s} \end{aligned} \tag{2.14}$$

This has one lead term and one washout term. Comparing the frequency response of the original electrical transfer function $g_{22}(s)$ with the compensated electrical transfer function $g_{22}(s)+p(s)g_{12}(s)$, the effect of the inverted notch characteristic of the PSS is to dominate the "switchback" characteristic and by doing so, alleviates its effect on the system. This approach is effective at different loading conditions. Thus the system, with the PSS in place, can be treated as a pseudo SISO system loop where the governor loop is first closed and the AVR-excitation loop is treated as a SISO system for the sole purpose of providing voltage disturbance rejection.

The use of QFT as a viable tool for designing power system stabilisers was gaining momentum, and Boje and Jennings [6] presented a tutorial on how such a design should be undertaken. A third order model was used and relates the inputs to the outputs as shown in Equation 2.15. The tutorial was used to examine existing PSS design for a system with the AVR and speed governor already in place.

$$\begin{aligned}\begin{bmatrix}\Delta\omega(s)\\ \hline \Delta v_t(s)\end{bmatrix} &= \boldsymbol{P}\begin{bmatrix}\Delta T_m(s)\\ \hline \Delta E_{fd}(s)\end{bmatrix}\\ \boldsymbol{P} &= \frac{1}{\Delta s}\begin{bmatrix}\tilde{p}_{11} & \tilde{p}_{12}\\ \tilde{p}_{21} & \tilde{p}_{22}\end{bmatrix}\end{aligned} \tag{2.15}$$

The design was done with decoupled SISO speed and voltage loops. The specification was twofold:

1. to guarantee robust stability for all likely operating conditions, and

2. to improve the performance of the system in a structured manner.

The performance can be specified as tracking performance and regulation performance. Details of the derivation of these specifications are found in the paper. It is important to note, however, that the behaviour of the system when meeting these performance specifications is significantly influenced by shaping the loop transmission function. The design procedure is summarised as follows

1. draw the locus of the plant elements as the plant parameters vary. This will result in plant templates,

2. choose a nominal fixed plant to use as a handle on the design process. Such a handle is used to generate nominal boundaries, which meet the design specifications, and

3. design the controller to satisfy the nominal boundaries at the discrete frequency points.

The design specifications are used as an initial starting point. The QFT method offers insight for design improvement, and this means that the resulting trade-offs can be clearly seen during the design procedure.

2.5 1999

A useful insight into how power system stabilisers are designed and tuned in industry is found in the paper by Lakmeeharan and Coker [23]. The study was to damp electromechanical oscillations, which might be induced on the ESKOM system by interconnections within the South African Power Pool. The phase lead required was determined by measuring the phase response from the voltage reference to the airgap torque. No mention is made of the operating condition at which the measurement was made. Of interest is the way the PSS gain was determined. Two methods were given. One method used the root locus to determine the gain that will result in maximum damping. The other method sets the initial value of the gain equal to the inverse of the compensator's time constant. Values above and below this are then tried to see which value would give the best performance.

Clearly a quantitative method is required which will allow convergence to a compensator that meets the design specifications.

Boje, Nwokah, and Jennings [7] describe such a quantitative method in detail. The design is approached from a control engineering rather than a power system viewpoint. Historically, PSS have been designed after the speed and terminal voltage loops have been closed. Boje *et al* suggest designing a forward path decoupler (to reduce the interaction between the speed and voltage) before the loops are closed, and then performing SISO QFT designs on the speed and voltage loops. The reason for this becomes apparent when it is realised that the initial control loops may destabilise the system before the addition of a PSS. Fadlalmoula [12] showed that one of the reasons a PSS is applied, is to reduce to the coupling between the speed and voltage loops. Boje, Nwokah expand on this fact by using the Perron root R as a measure of the interaction between the loops to design a PSS that guarantees robust stability.

The resulting QFT stability specification is of the form

$$\left|\frac{a(j\omega) + g_{PSS}b(j\omega)}{c(j\omega) + g_{PSS}(j\omega)d(j\omega)}\right| \leq R(\omega) \tag{2.16}$$

The i_{th} individual loop sensitivity is given by

$$S_i = \frac{1}{1 + l*} \tag{2.17}$$

If $|S_1 S_2 R| \leq 1$, then stability for the individual design of the torque-speed and field-terminal voltage loops will result in a stable design. If $\sqrt{|S_1 S_2 R|} \leq 0.5$, then the two loops will not be strongly coupled even when the loop gains are low.

The speed and voltage loops are then designed using SISO QFT methods

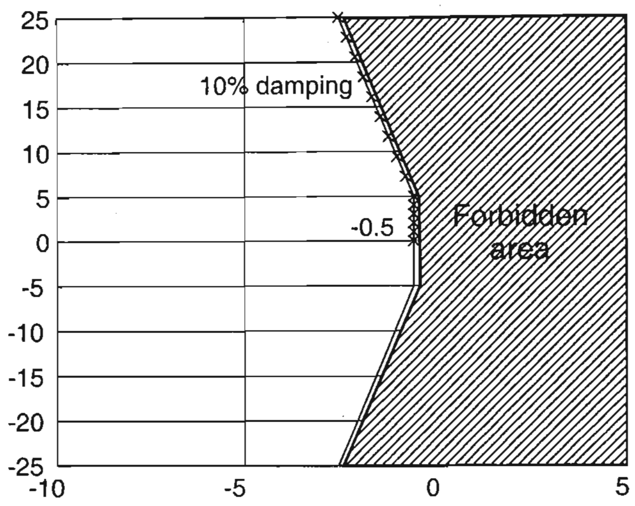


Figure 2.6: The D contour showing $z = 10\%$ and $a = 0.5$.

such that the controllers do not destabilise the existing system.

Rao and Sen [34] showed the application of QFT to PSS design from the power system perspective. The method used differed slightly from the one proposed by Horowitz. Instead of achieving robust stability of the closed loop with discrete complex frequency points on the imaginary axis, the closed loop poles are now chosen to lie within a modified D contour (Figure 2.6). The D contour is determined from the damping factor and the constraint that the real part of the dominant closed loop rotor eigenvalues is less than some value, a. If the designed PSS places the closed loop poles to the left of the contour for the given range of operating conditions, then the system achieves robust stability, and the damping requirements are met.

The closed loop will be robustly stable for all plants G of the plant set $\mathcal{G}$ if

1. the templates of the compensated plants $K(s)G(s)$ do not contain the point $(-180°, 0dB)$ on the Nichols chart for all $\omega \in \mathcal{R}$, and

2. the nominal closed loop is stable.

The procedure for designing the robust controller is

1. choose a set of points on the D contour for which robust stability is desired,

2. compute the plant templates as G is varied over $\mathcal{G}$ for each of the chosen point, and

3. choose a controller for which the nominal closed loop is stable within the D contour.

Shaping of the nominal loop transmission functions given by Horowitz [14] is impractical in this case. Rao and Sen suggest determining the controller coefficients by parameter optimisation. If the given form of the PSS transfer function is

$$G_{PSS} = K_s \frac{(1 + \tau_1 s)}{(1 + \tau_2 s)} \tag{2.18}$$

which is the most popular form used in industry, then the optimisation problem is to find the vector of controller parameters $[K_s, \tau_1, \tau_2]$ such that $g_i < 0, \quad i = 1, 2, \ldots n+2$. g_i is the minimum distance of the i_{th} compensated template from the $(-180°, 0dB)$ point on the Nichols chart and is negative if the critical point lies inside the template.

2.6 Summary

This chapter has highlighted the electromechanical instability, which occurs when trying to operate a generator close to its maximum power transfer capability. It introduced the concept of a power system stabiliser, which is generally used to damp out the electromechanical oscillations. More important, it showed the progress and development of the control theory and PSS implementation over the years. The most notable aspects include

1. identifying the electromechanical instability (DeMello and Concordia [10]),

2. analysing the power system instability (Larsen and Swann [26]),

3. implementing power system stabilisers in industry using various methods (Hoa and Agee [44]),

4. the realisation that a PSS decouples the speed and terminal voltage control loops (Fadlalmoula [12]), and

5. the recent application of quantitative feedback theory to the tuning of power system stabilisers (Boje, Nwokah, and Jennings [6], Rao and Sen [34]).

Chapter 3

Mathematical Modelling

> I have tried to avoid long computations, thereby following Riemann's postulate that proofs should be given through ideas and not voluminous computations.
>
> DAVID HILBERT
> *Report on Number Theory*

Most control engineering designs require a model of the system upon which the design is to be done. The three most common forms of such a model is a mathematical description either in state space, linear transfer function, or differential equations. The suitability of each format is dependent upon the particular application. In this thesis, the model is described using differential equations, and is then cast into state space form for ease of use with linear control theory.

The system in this case refers to a single machine infinite bus power system. It consists of a synchronous generator, the transmission line, the exciter, the automatic voltage regulator (AVR), the speed governor, and the infinite bus. In the frequency domain, the order is not a problem. The complexity of the final model is determined by how accurately the system is described. It was desired that the mathematical model closely resemble an existing system. A 7th order generator model resulted from the inclusion of damper windings. The exciter and AVR resulted in a 3rd order model. The governor was modelled as a 2nd order system while the transmission line was modelled as a static network. The modelling of fast transients using singular perturbation is a possibility, although it is not discussed in this thesis.

It was necessary to choose a frame of reference for modelling the power system. The choice of reference frame depends on the condition being modelled, although all reference frames will give the same end result. The reason for choosing a particular reference frame is that it will make the resulting computations easier and give a better insight into the nature of the condition.

3.1 The *d-q* transformation

The *d-q* transformation is particularly suited to modelling the power system. By placing the reference frame on the rotor of the synchronous machine and applying Park's transformation to the 3-phase differential equations, a set of reduced equations are obtained that eliminate the sinusoidal component from the original equations. Such changes in variables are not restricted to ac machinery, they are also applied to static, constant power devices. In fact, variables associated with transformers, transmission lines, capacitor banks, static var units, and loads are often represented in a frame of reference rotating at synchronous speed, which does not necessarily equal the rotor reference frame.

A change of variables which results in a transformation of the 3-phase variables of stationary circuit elements to the *d-q* reference frame is given by

$$\boldsymbol{f_{qd0}} = \boldsymbol{K_s f_{abc}} \tag{3.1}$$

where

$$\begin{aligned}
\boldsymbol{f_{qd0}} &= [f_q \quad f_d \quad f_0]^T \\
\boldsymbol{f_{abc}} &= [f_a \quad f_b \quad f_c]^T \\
\boldsymbol{K_s} &= \begin{bmatrix} \cos\theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \cos\theta & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \\
\theta &= \int_0^t \omega(\xi)\mathrm{d}\xi + \theta(0)
\end{aligned} \tag{3.2}$$

f can represent either voltage, current, flux linkage, or electric charge. The angular displacement θ must be continuous. The transformation of the three most common circuit elements will be considered. Transformation of a balanced 3-phase set is also considered, as it will be used later in the simulations.

3.1.1 Resistive elements

Given a 3-phase circuit equation of the form

$$\boldsymbol{v_{abc}} = \boldsymbol{R i_{abc}} \tag{3.3}$$

the *d-q* transformation (from Equation 3.1) is

$$\boldsymbol{v_{qd0}} = \boldsymbol{K_s R K_s^{-1} i_{qd0}} \tag{3.4}$$

Assuming the three phase resistance are equal, then $\boldsymbol{R}$ is a non zero, diagonal matrix with equal diagonal entries. Therefore

$$\boldsymbol{K_s R K_s^{-1}} = \boldsymbol{R} \tag{3.5}$$

It can be seen that the resistance is not affected by the transformation from 3-phase to d-q. If the phase resistances are unbalanced, then the resistance matrix will contain sinusoidal functions of θ except when $\omega = 0$. When $\omega = 0$, $\boldsymbol{K_s}$ is purely algebraic.

3.1.2 Inductive elements

The equation describing a simple 3-phase circuit with an inductance is

$$\boldsymbol{v_{abc}} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\lambda_{abc}} \tag{3.6}$$

In terms of the d-q transformation

$$\begin{aligned} \boldsymbol{v_{qd0}} =& \boldsymbol{K_s}\frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{K_s^{-1}\lambda_{qd0}}\right] \\ =& \boldsymbol{K_s}\frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{K_s^{-1}}\right]\boldsymbol{\lambda_{qd0}} + \boldsymbol{K_s K_s^{-1}}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\lambda_{qd0}} \end{aligned} \tag{3.7}$$

It can be shown (after some work, and the use of trigonometric identities) that

$$\boldsymbol{K_s}\frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{K_s^{-1}}\right] = \omega \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.8}$$

Therefore, Equation 3.7 may be expressed as

$$\boldsymbol{v_{qd0}} = \omega \begin{bmatrix} \lambda_d \\ -\lambda_q \\ 0 \end{bmatrix} + \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \lambda_q \\ \lambda_d \\ 0 \end{bmatrix} \tag{3.9}$$

The terms with ω are commonly referred to as "speed voltage".

3.1.3 Capacitive elements

The current in a 3-phase circuit with capacitors is

$$\boldsymbol{i_{abc}} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{q_{abc}} \tag{3.10}$$

In terms of the substitute variables

$$\begin{aligned} \boldsymbol{i_{qd0}} =& \boldsymbol{K_s}\frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{K_s^{-1} q_{qd0}}\right] \\ =& \boldsymbol{K_s}\frac{\mathrm{d}}{\mathrm{d}t}\left[\boldsymbol{K_s^{-1}}\right]\boldsymbol{q_{qd0}} + \boldsymbol{K_s K_s^{-1}}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{q_{qd0}} \end{aligned} \tag{3.11}$$

Using Equation 3.8 to simplify Equation 3.11, the resulting expression is similar to Equation 3.9.

$$i_{qd0} = \omega \begin{bmatrix} q_d \\ q_q \\ 0 \end{bmatrix} + \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} q_q \\ q_d \\ 0 \end{bmatrix} \tag{3.12}$$

3.1.4 Three phase balanced set transformation

A balanced set is defined as a set of equal amplitude sinusoidal quantities that are displaced by 120°. Given the following balanced set

$$\begin{aligned} f_a =& \sqrt{2} f \cos(\theta_{ef}) \\ f_b =& \sqrt{2} f \cos(\theta_{ef} - \frac{2\pi}{3}) \\ f_c =& \sqrt{2} f \cos(\theta_{ef} + \frac{2\pi}{3}) \\ \theta_{ef} =& \int_0^t \omega_e(\xi)\mathrm{d}\xi + \theta_{ef}(0) \end{aligned} \tag{3.13}$$

the resulting transformation is

$$\begin{aligned} f_q =& \sqrt{2} f \cos(\theta_{ef} - \theta) \\ f_d =& \sqrt{2} f \sin(\theta_{ef} - \theta) \\ f_0 =& 0 \end{aligned} \tag{3.14}$$

It is important to distinguish between θ_{ef} and θ. The angular position of the synchronously rotating frame is θ. The angular position of each electrical variable is θ_{ef}.

3.2 Mathematical model of a synchronous generator

The equations of a 7th order synchronous generator model are presented in this section. A derivation of these equations can be found in [20]. Also, most standard texts [21], [41], and [1] offer in-depth analysis of the synchronous machine.

For the purpose of this study, and for compatibility reasons, the synchronous machine's equations will be stated in terms of flux linkages. The stator variables are transformed into the rotor reference frame. The equations may also be written in compact matrix format using the current as the state variable. The equations describe a machine with 3 *d-axis* windings and 2 *q-axis* windings. This is the popular 2:1 Philips-Heffron model, illustrated in Figure 3.1.

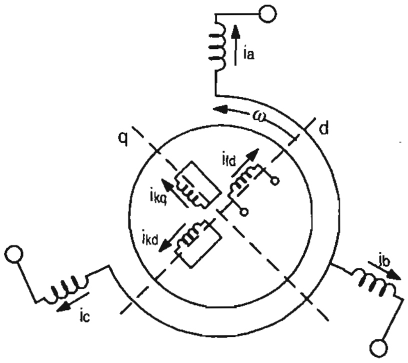


Figure 3.1: Physical representation of a synchronous machine showing the 3-phase stator windings and the *d*-*q* rotor windings.

3.2.1 The 5 electrical equations

The synchronous machine is usually operated as a generator, so it is convenient to assume that the direction of positive stator current is out of the terminals as shown in Figure 3.1. Using this convention, the voltages in the 3-phase stator windings are

$$\boldsymbol{v_{abc}} = -\boldsymbol{R_s i_{abc}} + \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\lambda_{abc}} \tag{3.15}$$

Writing Equation 3.15 in *d*-*q* variables results in

$$\boldsymbol{v_{sqd}} = -\boldsymbol{R_s i_{sqd}} + \omega\boldsymbol{\lambda_{sqd}} + \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\lambda_{sqd}} \tag{3.16}$$

where

$$\boldsymbol{\lambda_{sqd}} = [\lambda_{sd} \quad -\lambda_{sq} \quad 0]^T \tag{3.17}$$

The s subscript denotes a stator quantity. The restriction on Equation 3.16 is that the diagonal resistance matrix $\boldsymbol{R_s}$ must have equal values for the stator resistances.

The rotor voltages are given by

$$\boldsymbol{v_{rqd}} = \boldsymbol{R_r i_{rqd}} + \omega\boldsymbol{\lambda_{rqd}} + \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\lambda_{rqd}} \tag{3.18}$$

It would be useful to have the electrical equations in terms of inductances, rather than flux linkage. To convert from flux linkage to inductances in the *d*-*q* reference frame, the flux linkage for a system with a linear magnetic circuit is first expressed as

$$\boldsymbol{\lambda_{abc}} = \boldsymbol{L_s i_{abc}} \tag{3.19}$$

Applying the transformation

$$\boldsymbol{\lambda}_{qd} = \boldsymbol{K}_s \boldsymbol{L}_s \boldsymbol{K}_s^{-1} \boldsymbol{i}_{qd} \tag{3.20}$$

Using this result to express the flux linkages of the synchronous machine in terms of inductances gives

$$\begin{bmatrix} \boldsymbol{\lambda}_{sqd} \\ \boldsymbol{\lambda}_{rqd} \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_s \boldsymbol{L}_s \boldsymbol{K}_s^{-1} & \boldsymbol{K}_s \boldsymbol{L}_r \\ \frac{2}{3} \boldsymbol{L}_r \boldsymbol{K}_s^{-1} & \boldsymbol{L}_r \end{bmatrix} \begin{bmatrix} -\boldsymbol{i}_{sqd} \\ \boldsymbol{i}_{rqd} \end{bmatrix} \tag{3.21}$$

$\boldsymbol{L}_s$ and $\boldsymbol{L}_r$ denote the matrix inductance (leakage and mutual) on the stator and rotor respectively. The $\frac{2}{3}$ factor comes about when the rotor windings are referred to the stator. It is sufficient to state the elements of the inductance matrix in Equation 3.21 in expanded form. Derivation of the terms can be found in [8]. The expanded elements of Equation 3.21 are

$$\boldsymbol{K}_s \boldsymbol{L}_s \boldsymbol{K}_s^{-1} = \begin{bmatrix} L_l + L_{mq} & 0 & 0 \\ 0 & L_l + L_{md} & 0 \\ 0 & 0 & L_l \end{bmatrix} \tag{3.22}$$

$$\boldsymbol{K}_s \boldsymbol{L}_r = \begin{bmatrix} L_{mq} & L_{mq} & 0 \\ 0 & L_{md} & L_{md} \\ 0 & 0 & 0 \end{bmatrix} \tag{3.23}$$

$$\frac{2}{3} \boldsymbol{L}_r \boldsymbol{K}_s^{-1} = \begin{bmatrix} L_{mq} & 0 & 0 \\ L_{mq} & L_{md} & 0 \\ 0 & L_{md} & 0 \end{bmatrix} \tag{3.24}$$

It is convenient to write Equation 3.16 and Equation 3.18 in expanded format as it allows for a much clearer understanding of the different voltage contributions. Subscripts f and k denote the field and damper circuits respectively. The five differential voltage equations that describe the synchronous machine are

1. *d-axis*:

$$v_d = -R_s i_d + \omega \lambda_q + \frac{\mathrm{d}}{\mathrm{d}t} \lambda_d \tag{3.25}$$

$$v_{fd} = R_{fd} i_{fd} + \frac{\mathrm{d}}{\mathrm{d}t} \lambda_{fd} \tag{3.26}$$

$$v_{kd} = R_{kd} i_{kd} + \frac{\mathrm{d}}{\mathrm{d}t} \lambda_{kd} \tag{3.27}$$

2. *q-axis*:

$$v_q = -R_s i_q + \omega \lambda_d + \frac{\mathrm{d}}{\mathrm{d}t} \lambda_q \tag{3.28}$$

$$v_{kq} = R_{kq} i_{kq} + \frac{\mathrm{d}}{\mathrm{d}t} \lambda_{kq} \tag{3.29}$$

The algebraic equations that relate the flux linkages to the currents are

1. *d-axis*:

$$\lambda_d = -L_l i_d + L_{md}(-i_d + i_{fd} + i_{kd}) \tag{3.30}$$
$$\lambda_{fd} = L_{lfd} i_{fd} + L_{md}(-i_d + i_{fd} + i_{kd}) \tag{3.31}$$
$$\lambda_{kd} = L_{lkd} i_{kd} + L_{md}(-i_d + i_{fd} + i_{kd}) \tag{3.32}$$

2. *q-axis*:

$$\lambda_q = -L_l i_q + L_{mq}(-i_q + i_{kq}) \tag{3.33}$$
$$\lambda_{kq} = L_{lkq} i_{kq} + L_{mq}(-i_q + i_{kq}) \tag{3.34}$$

Equations 3.25–3.34 describe the equivalent circuits representing the synchronous machine electrical system in Figures 3.2–3.4.

3.2.2 The 2 mechanical equations

The derivation of the electrical torque for the synchronous machine is complicated, and will not be shown here. An alternative method for determining the electrical torque [36] is given. Reference [20] provides a derivation from the fundamental machine parameters.

The electrical power at the generator terminals is given by

$$P_e = v_q i_q + v_d i_d \tag{3.35}$$

Substituting for v_q and v_d, the expression becomes

$$P_e = \left(\frac{\mathrm{d}}{\mathrm{d}t}\lambda_q\right) i_q + \left(\frac{\mathrm{d}}{\mathrm{d}t}\lambda_d\right) i_d + (\lambda_d i_q - \lambda_q i_d)\omega + R_s(i_q^2 + i_d^2) \tag{3.36}$$

Looking at the three terms that make up P_e:

1. $\frac{\mathrm{d}}{\mathrm{d}t}\lambda_q i_q + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_d i_d$ is the rate of change of magnetic energy with respect to time.

2. $(\lambda_d i_q - \lambda_q i_d)\omega$ represents the power transferred across the air-gap from the rotor to the stator.

3. $R_s(i_q^2 + i_d^2)$ is the stator resistance loss.

The electrical torque is defined as

$$T_e = \frac{airgap\ \ power}{rotor\ \ speed} \tag{3.37}$$

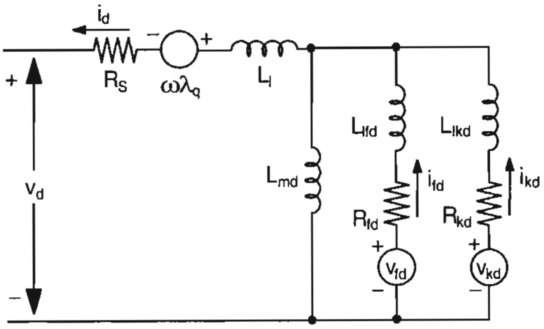


Figure 3.2: Equivalent circuit representation of *d-axis* electrical equations.

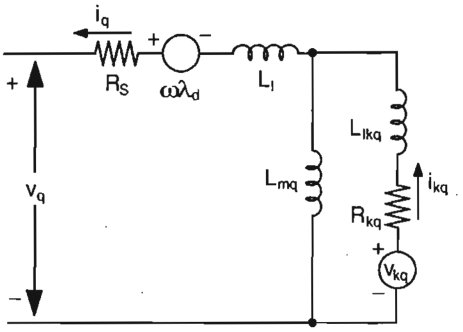


Figure 3.3: Equivalent circuit representation of *q-axis* electrical equations

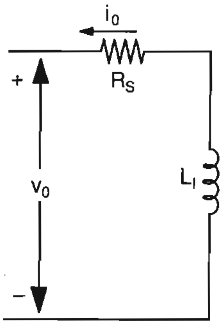


Figure 3.4: Equivalent circuit representation of the 0-component of the electrical equations. Usually this component equals zero.

Substituting the 2nd term from P_e in Equation 3.37 gives

$$\begin{aligned} T_e =& \frac{(\lambda_d i_q - \lambda_q i_d)\omega}{\omega} \\ =& \lambda_d i_q - \lambda_q i_d \end{aligned} \tag{3.38}$$

The electrical torque and rotor speed are related by

$$T_e = -J \frac{2}{\mathrm{P}} \frac{\mathrm{d}}{\mathrm{d}t} \omega + T_m \tag{3.39}$$

where J is the inertia of the generator expressed in kilograms · meters^2 (kg · m^2), P is the number of poles and T_m is the mechanical torque applied to the generator shaft. T_m is positive, in keeping with the sign convention given in Appendix A.1.

The two mechanical equations are therefore

$$T_e = \lambda_d i_q - \lambda_q i_d \tag{3.40}$$

and including the damping and viscous coefficients, D and K respectively

$$\frac{\mathrm{d}}{\mathrm{d}t} \omega = \frac{\mathrm{P}}{2J} (T_m - T_e - D\omega - K\delta) \tag{3.41}$$

3.3 The transmission line

The transmission line is modelled as a lumped parameter model consisting of an inductance in series with a resistance. The lumped parameter model is accurate for line lengths less than 200km [20]. The electrical diagram representing the transmission line is shown in Figure 3.5. The transmission line needs to be transformed into the *d*-*q* reference frame so that it can be modelled in conjunction with the synchronous generator. Each phase voltage in Figure 3.5 can be expressed as the sum of voltages across each element i.e.

$$\begin{aligned} v_a =& v_{aR} + v_{aL} \\ v_b =& v_{bR} + v_{bL} \\ v_c =& v_{cR} + v_{cL} \end{aligned} \tag{3.42}$$

Transforming Equation 3.42 to the *d*-*q* reference frame gives

$$\begin{aligned} v_q =& v_{qR} + v_{qL} \\ v_d =& v_{dR} + v_{dL} \\ v_0 =& v_{0R} + v_{0L} \end{aligned} \tag{3.43}$$

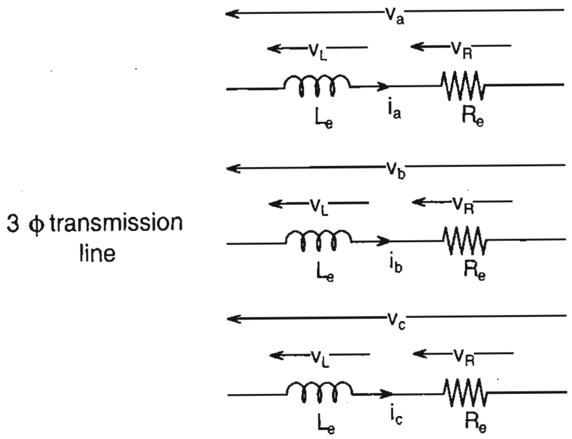


Figure 3.5: Circuit diagram of a 3-phase transmission line modelled as lumped parameters.

From Equation 3.43 the voltage drops across the resistances can be expressed in terms of the current

$$\begin{aligned} v_{qR} =& R_e i_q \\ v_{dR} =& R_e i_d \\ v_{0R} =& R_e i_0 \end{aligned} \tag{3.44}$$

Using Equation 3.9 as a guide for inductive elements, the voltage drops across the inductances can be written as

$$\begin{aligned} v_{qL} =& \omega\lambda_d + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_q \\ v_{dL} =& -\omega\lambda_q + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_d \\ v_{qL} =& \frac{\mathrm{d}}{\mathrm{d}t}\lambda_0 \end{aligned} \tag{3.45}$$

From Equation 3.19 and Equation 3.20, the flux linkages in Equation 3.45 can be written in terms of inductances as

$$\begin{aligned} \lambda_q =& (L_e - M)i_q \\ \lambda_d =& (L_e - M)i_d \\ \lambda_0 =& (L_e + 2M)i_0 \end{aligned} \tag{3.46}$$

M is the mutual flux linking the transmission lines. Thus, from Equation 3.44 and Equation 3.45 the final equations describing a transmission line in *d*-*q* variables are

$$\begin{aligned} v_q &= R_e i_q + \omega\lambda_d + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_q \\ v_d &= R_e i_d + \omega\lambda_q + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_d \\ v_0 &= R_e i_0 + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_0 \end{aligned} \tag{3.47}$$

In most cases the v_0 voltage component is zero, and is not included in the model equations.

3.4 Summary

This chapter has developed the mathematical models that represent the elements of a typical power system. It is now possible to simulate the power system under various conditions using these equations.

The models are detailed enough to allow accurate representation of actual systems. Most often, a third order model of the generator is used, with the exciter represented by a simple first order system, and the governor dynamics are ignored. This is useful for insight and clarity when doing a design.

Understanding the *d*-*q* transformation lays the foundation for understanding and deriving the system equations. However, there are cases where the *d*-*q* transformation is not suitable. Examples of these are when there are power electronic devices to model. In this case, the devices are best modelled in three phase, and the output quantities transformed into *d*-*q* coordinates. This is commonly known as a hybrid system.

Chapter 4

Matlab Power System Blockset

> God does not care about our mathematical difficulties. He integrates empirically.
>
> ALBERT EINSTEIN (1879–1955)
> *Quest*

Computer modelling and analysis of electric power systems is easier and faster today than it was a few years ago. The focus now shifts from making sure all equations are correct and in the right place to a global overview of the system and the effect of design on the system. This reduction in computation effort is accomplished by the Matlab Power System Blockset [17], or PSB. There are other commercial software packages/applications specifically designed for simulating power systems such as *PSCAD* and *EMTDC*.

The PSB is an integrated set of Simulink [31] blocks and tools that allow for quick modelling of power system components. An entire power system, including the power electronics, can be built and simulated, and the results compared against a simulation of a linear power system model. The PSB also allows for nonlinear simulation of synchronous generators, distributed parameter transmission lines, and induction machinery.

The use of the PSB is illustrated by modelling and simulating an existing single machine infinite bus system.

4.1 Inside the PSB

The libraries contain models of typical power system components. These models are proven ones and are obtained from textbooks. Their validity is based on the experience of the Power Systems Testing and Simulation

Laboratory of Hydro-Quebec [17].

Models are built up using the electrical blocks from the Simulink Power System Blockset library. Once the model has been build, the simulation process can start. This process consists of the following:

4.1.1 Parse network topology

In this process, the blocks in the model file are separated into Simulink blocks and Power System blocks. The Simulink blocks are used to solve the linear part of the simulation. The nonlinear part of the simulation is made up of the Power System blocks. The network topology is converted into a node information file that is used in the next step.

4.1.2 Obtain state-space model

The state space model is built by reading the node information and the `RLC` (resistor, inductance, and capacitor) matrix of the electrical network to which the machine is connected. This is one of many parameter matrices. From this information, the linear state space model representing the network is obtained. Other parameter matrices are the `SOURCE` matrix, which holds information about all the voltage and current sources in the model, the `SWITCH` matrix, and the `LINE_DIST` matrix that represents the distributed parameter line. The inputs to the state space model are currents and the output is voltage. The state variables are voltages.

4.1.3 Build Simulink model

A private Simulink model is built and stored in one of the voltage or current measurement blocks in the main PSB Simulink model. This is the reason why every PSB Simulink model must have at least one measurement block. Nonlinear devices are simulated as current sources in the feedback loop of the state space model. Therefore, the nonlinear models are designed to have voltages as inputs and currents as outputs. As such, they cannot be connected in series with inductors, and their terminals cannot be left open.

Figure 4.1 shows the generic private Simulink model. The connections from the PSB Simulink model inputs and outputs to the private Simulink model are made via standard Simulink "GOTO" blocks.

4.1.4 Start simulation

Once the initialisation process has been completed, the model is simulated. The private Simulink model is transparent, and it runs in the background. In fact, it is the simulation source model. The PSB model is just a user-friendly graphical user interface, and is merely a tool for drawing the model

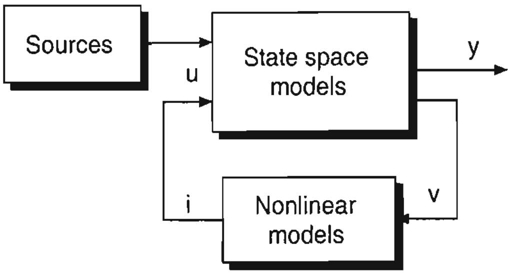


Figure 4.1: Generic private Simulink model built during simulation.

in a intuitive manner, and displaying the results in a format that is easy to understand.

4.1.5 The nonlinear synchronous generator model

It is important to understand how the PSB models the synchronous generator, as this model will be used for nonlinear simulation. In addition, the linearised model for QFT design will be derived from this nonlinear model.

The model is derived using fluxes as the state variables. The electrical part of the machine is represented as a fifth order state space model. The equations describing electrical part of the machine are the same as Equations 2.24–2.33. Therefore, compatibility is maintained with the PSB model and the linear model that is derived from these equations.

The mechanical part of the machine is modelled as

$$\begin{aligned} \dot{\omega} &= \frac{1}{2H}\left(T_m - T_e - D\omega\right) \\ \dot{\delta} &= \omega \end{aligned} \tag{4.1}$$

This equation has the damping constant D included, and per unit quantities are used instead of actual values. The per unit inertia H is obtained from the inertia constant J. Derivation of this relationship can be found in [33].

$$H = \frac{2J(\omega_{base})^2}{\mathrm{P}^2(P_{base})} \tag{4.2}$$

where P_{base} is the machine power base value and ω_{base} is the speed base value.

4.2 Simulating a SMIB

To illustrate the use of the PSB, a single synchronous machine connected to an infinite bus will be considered. Information on how to enter parameters

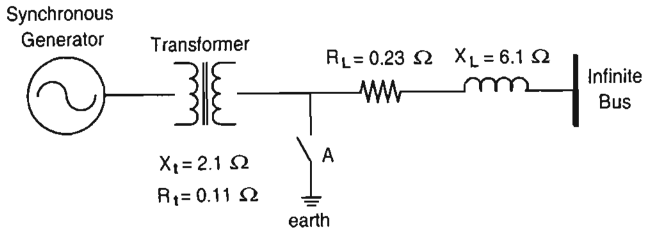


Figure 4.2: The power system to be modelled.

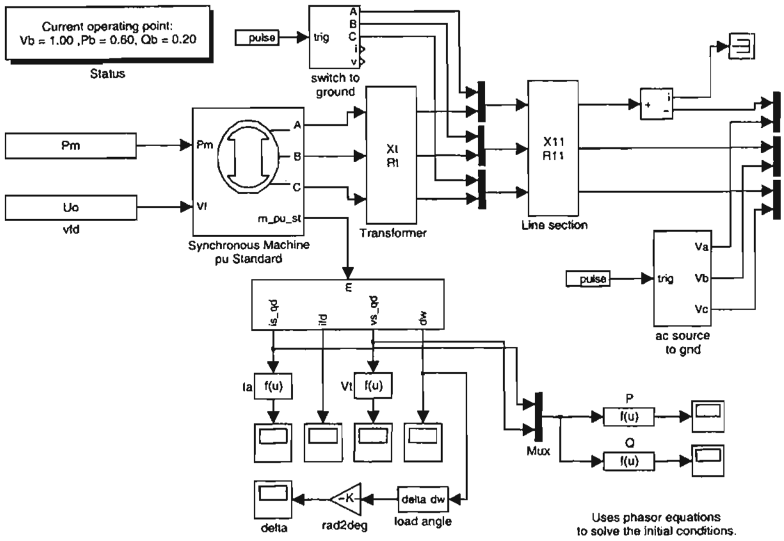


Figure 4.3: The equivalent Simulink model.

for the synchronous generator model, and other components, can be found in [17]. The circuit shown in Figure 4.2 is to be set up. This circuit can be broken down into components found in the PSB. To simulate an infinite bus, voltage sources are used. Figure 4.3 shows the PSB model. It is important to set the simulation parameters properly. By default, Matlab [29] uses a ode45 (Runga Kutta 4/5) variable step solver. This is inadequate, as the simulation will not converge unless a very small step size is used. Rather, a stiff solver should be used. The stiff solver is a multistep method and has better numerical stability properties on stiff, but smooth functions. In this simulation, the ode15 stiff solver was used. The machine parameters used in the simulation were taken from Micro-Alternator 2 (left hand side when facing the control panel). The advantage of using these parameters is that a rough idea of the synchronous micro machine behaviour can be gained. As can be seen from Figure 4.3, all components are found in the PSB library, with the exception of "Trig ABC" and "AC Source to Ground". These components are subset models and simulate the ground faults shown in Figure 4.2. "AC Source to Ground" contains the voltage sources necessary to simulate an infinite bus.

Variables `i_qd` and `v_qd` are sent to function blocks to calculate the armature current and terminal voltage respectively. These variables are also sent to a multiplexer, whose output is used to calculate the active and reactive power flow in the system.

The setpoints for the synchronous generator model were obtained by solving the steady state phasor equations. The active power was set at $P_b = 0.6$ p.u. and the reactive power was set at $Q_b = 0.2$ p.u. The derivation of initial conditions from such equations can be found in Appendix A. The maximum simulation step size was set to 0.01, and the relative tolerance was set to 0.001. The fifth order variable step mode was used. The reason for such tight tolerances is that the synchronous machine equations are nonlinear differential equations, which must be solved simultaneously. Small step sizes and tolerances mean longer simulation times, but it increases the chance of convergence. The program listing for this simulation can be found in Appendix F.1.

4.3 Simulation Results

The simulation was run with a simulated fault on point A as shown in Figure 4.2 at time $t = 5$ seconds. The duration of the fault was 300 milliseconds. Graphs of the armature current, terminal voltage, rotor angle, and active and reactive powers are shown.

The private Simulink model is shown in Figure 4.4. This was built during the initialisation stage. The nonlinear synchronous machine, along with the nonlinear switch model, is connected across the network state space matrix. The "Sources" contain the setpoint constants, and the infinite bus is modelled within the state space matrix. The simulation results are shown in Figure 4.5. Notice the transient disturbance that occurs at time $t = 0$. The initial conditions for this simulation model were deliberately modified by increasing them by 10%. This shows that incorrect initial conditions will result in a transient that dies off (if the system is stable) as the machine returns to steady state. How quickly the machine return to steady state is governed by the machine constants.

Figure 4.6 shows the simulation results with the correct initial conditions. By looking at the post-fault data, most noticeably the peaks, it can be seen that the initial conditions affect the accuracy of the simulation run. The post-fault load angle (and the first swing peak), for example, is greater in the incorrect simulation than in the correct simulation.

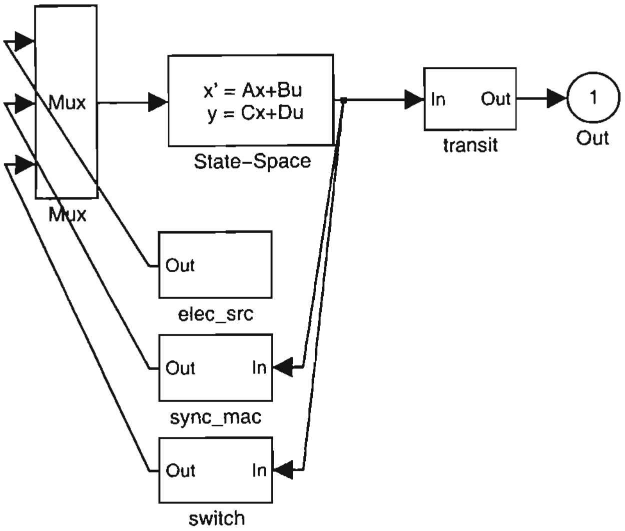


Figure 4.4: Private Simulink model built during simulation initialisation.

4.4 Summary

The Simulink model is applicable to both large signal, and small signal analysis, although only the large signal analysis was demonstrated. There is a lot more detail to the PSB than what was described here. The phase sequence of the voltage sources simulating the infinite bus is critical to the success of the simulation. The nonlinear effects of saturation, for which the PSB has made provisions, can be modelled as desired.

It was shown that the Simulink PSB uses models that are described by the equations in Chapter 2. However, linear control design methods cannot be undertaken, since these equations (and hence the PSB models themselves) are nonlinear differential equations. Of course, nonlinear control theory can be applied, but that is outside the scope of this thesis. Rather, the linear QFT method discussed in the next chapter will be used to design a controller. This controller can be used in the PSB simulation, and its effect observed. The PSB is a quick and intuitive tool for simulating large power systems. It provides accurate data and gives a general idea of how a power system will respond to certain disturbances.

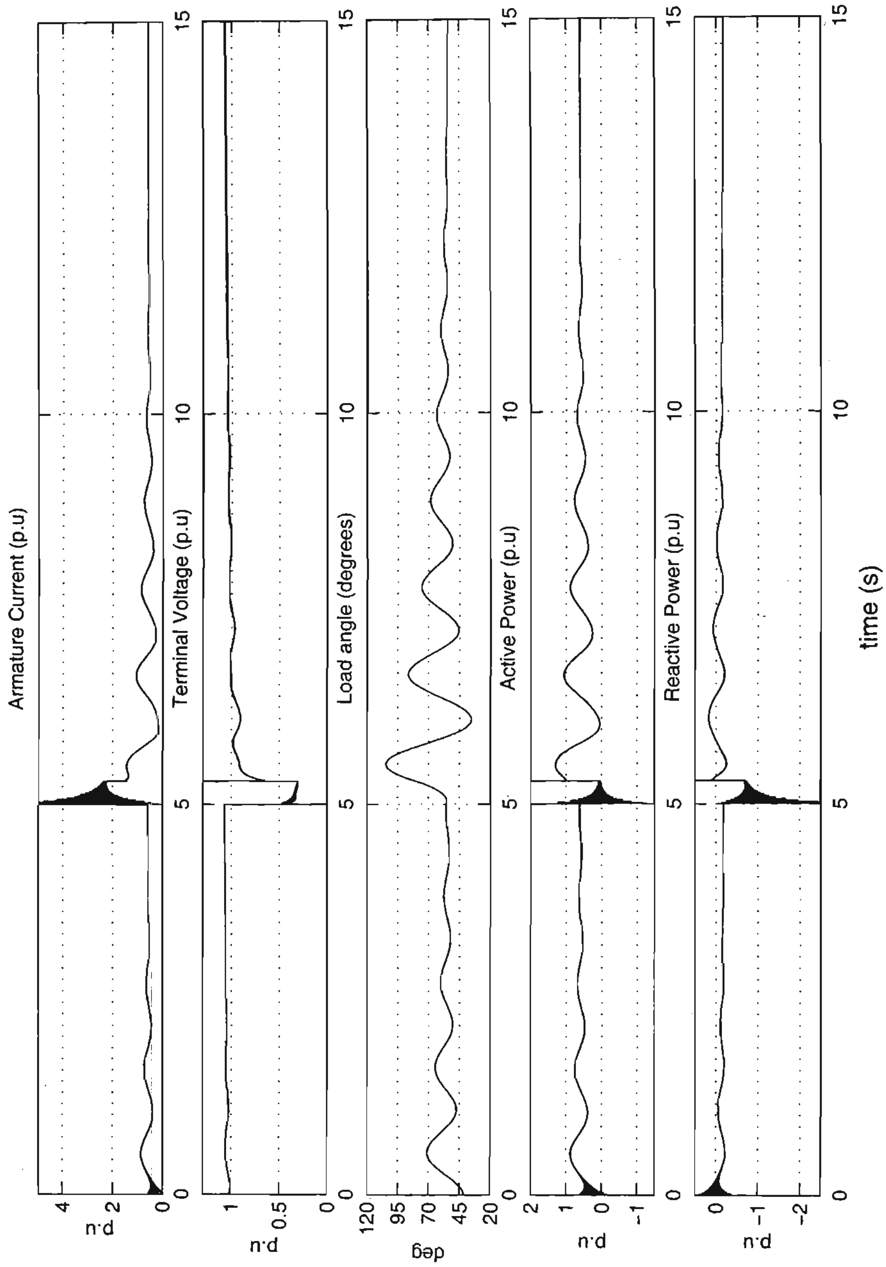


Figure 4.5: Simulation results with incorrect initial conditions.

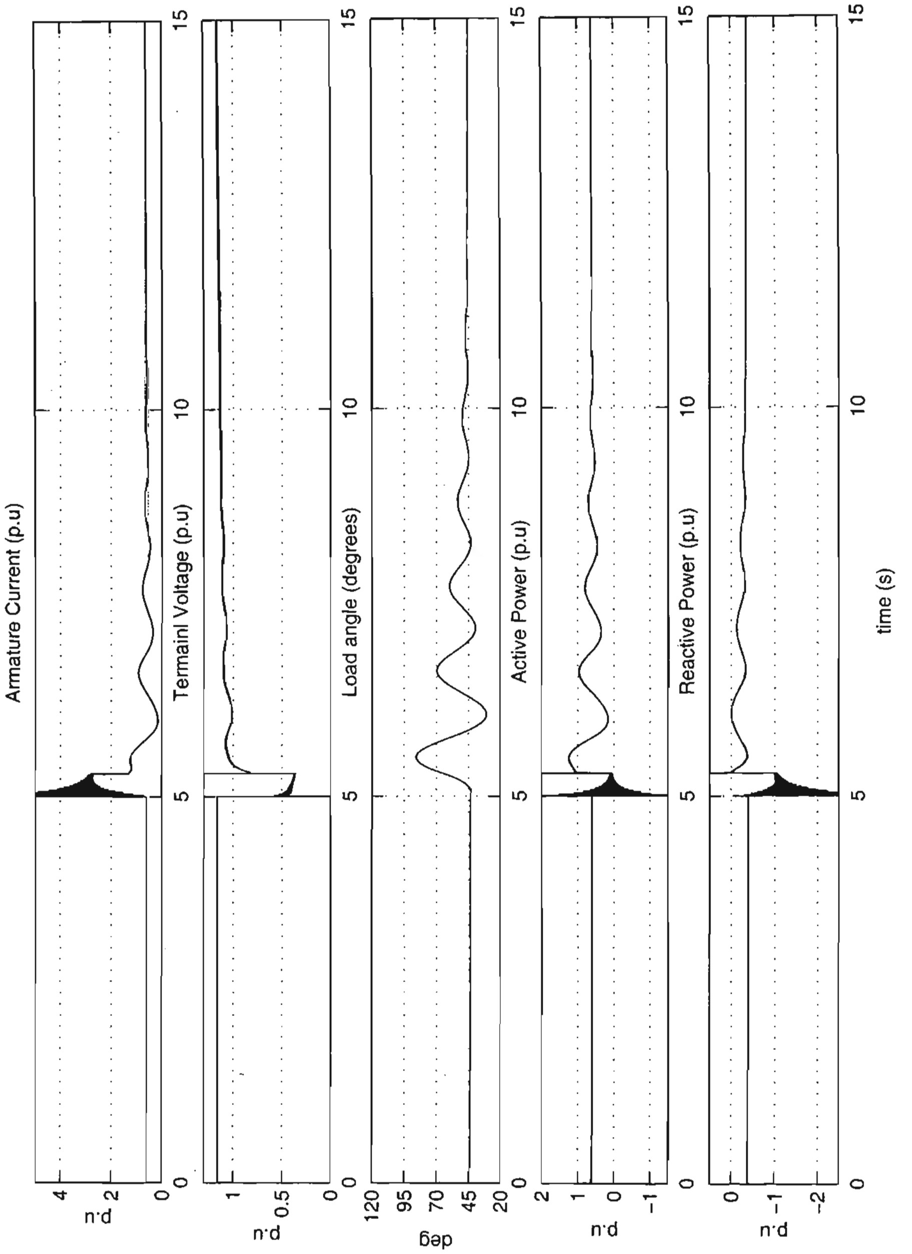


Figure 4.6: Same simulation, but with the correct initial conditions.

Chapter 5

Quantitative Feedback Theory

> Every new body of discovery is mathematical in form, because there is no other guidance we can have.
>
> CHARLES DARWIN
> *Mathematical Maxims & Minims*

QFT is a frequency domain method that was developed by Horowitz [14]. The Horowitz observation is that feedback is only necessary if there is uncertainty either of the dynamics or of the external signals (e.g. unmeasured disturbances). Otherwise, any behaviour can be achieved by open loop prefiltering. Early designs were based on the complex arithmetic plane. Later, it was discovered [15] that the log polar plot of magnitude vs. phase (Nichols Chart) was more convenient and easier to use.

The QFT design procedure is highly transparent because the stability and performance criteria are always visible during the design process and this makes it ideal for practical control design. Since QFT is based on the frequency response data, complex plants can be modelled. The reason for using QFT is that the design trade-offs at each step can be seen by the control designer, and corrections can be made accordingly.It is an iterative process, and also allows the designer to compare different designs. A system cannot always be defined exactly; there will always be tolerances either in parameter constants or operating conditions. For example, if a root locus design has poles close to the *$j\omega$-axis*, then small variations in the pole values may lead to instability. QFT was designed with the following considerations.

1. **State Space vs. Transfer Function**

   Transfer function representation was chosen because all states in the transfer function are observable. Since the transfer function exists, the system is controllable. Only the relevant states are available for processing. With non-minimal state-space there are the problems of observability and controllability. All the states must be calculated even if they are not used in the design process. Non-minimum phase as well as the "cost of feedback" [5] is not readily shown in state-space representation.

2. **Time Domain vs. Frequency Domain**

   Frequency domain was chosen because it eliminates solving the convolution integral which arises from the Laplace Transform. The complex s-domain does not offer the convenience that the frequency domain offers.

3. **Sensitivity Function vs. Loop Transmission**

   When doing a design, the bandwidth available becomes of primary interest, and the trade-off between bandwidth and system performance is the so-called "cost of feedback". The loop transmission is very sensitive to bandwidth and hence allows the designer to develop controllers to specification limits.

4. **Unstructured Uncertainty**

   This refers to the problem of dealing with the high frequency sensor noise or model uncertainty of plants. The solution is obviously to get the fastest roll-off of the loop transmission at the smallest frequency possible. Since the design is done in the frequency domain, the designer can immediately see how feasible this is.

   Specifications can be made in either the time domain or frequency domain. There has been very little research done on formulating time domain specifications [14]. However, time domain responses can be translated to frequency domain tolerances which lead to bounds on the loop transmission function. The frequency response completely specifies the transfer function, which in turn uniquely determines the step response [15]. Thus only the magnitude of the frequency response need be considered when creating a bound on the frequency domain. The phase is directly related to the magnitude via Bode Integrals [5] for minimum phase systems only. It is much more common to give the specifications in the frequency domain than in the time domain.

The beauty of QFT lies in the fact that design can be done based solely on frequency response measurements. An analytical model need not be formed. In fact, for some complex plants, there is no analytical solution. The trade-off, however, is that the closed-loop stability cannot be analysed via a computer. Analysis of the closed-loop stability is limited to the measured frequency range. The important fact, though, is that uncertainty has been included.

5.1 The Nichols Chart

If frequency domain is the domain of choice, then the Nichols Chart is the plane of choice. Since the Nichols Chart shows magnitude and phase, both the gain and phase margins can be easily seen.

The Nichols Chart has logarithmic magnitude on the *y-axis* and phase on the *x-axis*. Every complex number can be represented in terms of polar coordinates r and θ. r and θ are open-loop values, with r typically ranging from $-\infty dB$ to $+\infty dB$, and θ between 360° and 0°.

The Nichols Chart can be compared to other representations, and its advantages seen. Bode diagram representation shows magnitude vs. frequency and phase vs. frequency and offers a less transparent insight into the design process.

Nyquist diagram representations show a polar plot of magnitude and phase vs. frequency. Nyquist phase plots are usually greater than 360° which results in n windings around the origin. This is dealt with in the $r\theta$ plane by using multi-sheeted Nichols Charts.

A multi-sheeted Nichols Chart is simply a Nichols Chart whose phase axis has been extended. However, single-sheet Nichols Charts may also be used by making use of the complex conjugate property of complex numbers [9]. Consider the following transfer function,

$$L(s) = \frac{k}{(s+1)(s+5)(s+10)}\ , \quad k > 0 \tag{5.1}$$

Equation 5.1 is plotted on a Nichols Chart. Figures 5.1, 5.2, and 5.3 show the plot on a multi-sheeted, single-sheet, and half-sheet Nichols Chart respectively. The interior of a Nichols Chart usually has a grid representation of the closed-loop response of

$$\left|\frac{L(s)}{1+L(s)}\right|\ , \quad \angle\frac{L(s)}{1+L(s)} \tag{5.2}$$

The inverse Nichols Chart can also be formed by plotting grids of constant

$$\left|\frac{1}{1+L(s)}\right|\ , \quad \angle\frac{1}{1+L(s)} \tag{5.3}$$

In QFT, the closed-loop grid is used to form open-loop bounds i.e. the closed-loop specifications are translated into QFT bounds. The definition of stability on a Nichols Chart is derived from the Nyquist diagram. A crossing

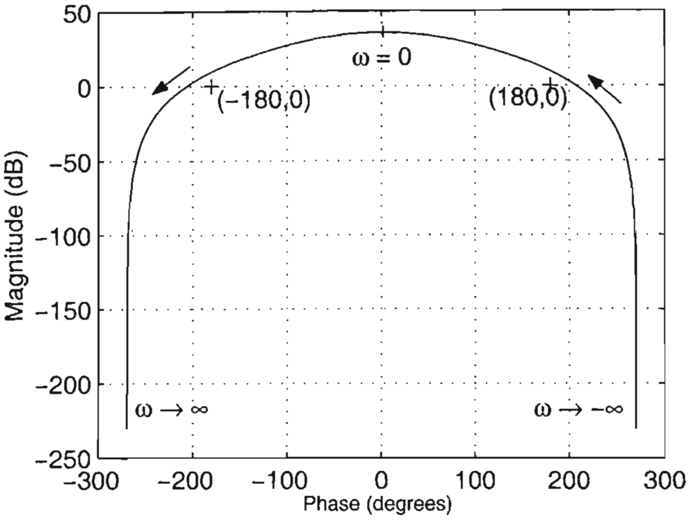


Figure 5.1: A plot on a multi-sheeted Nichols Chart.

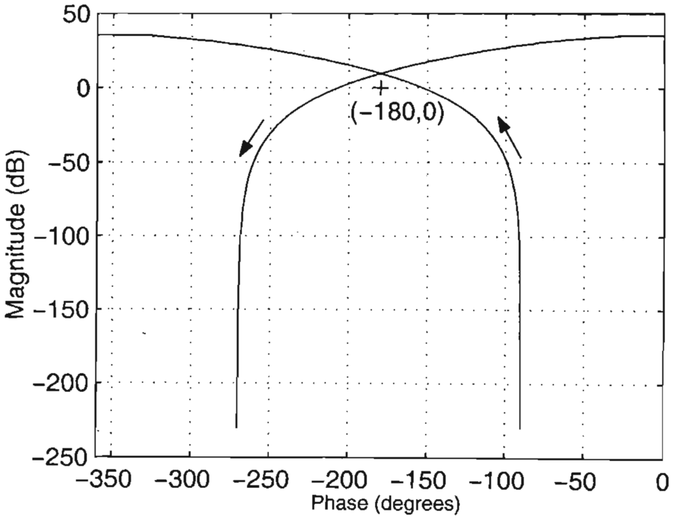


Figure 5.2: Same plot on a single-sheet Nichols Chart.

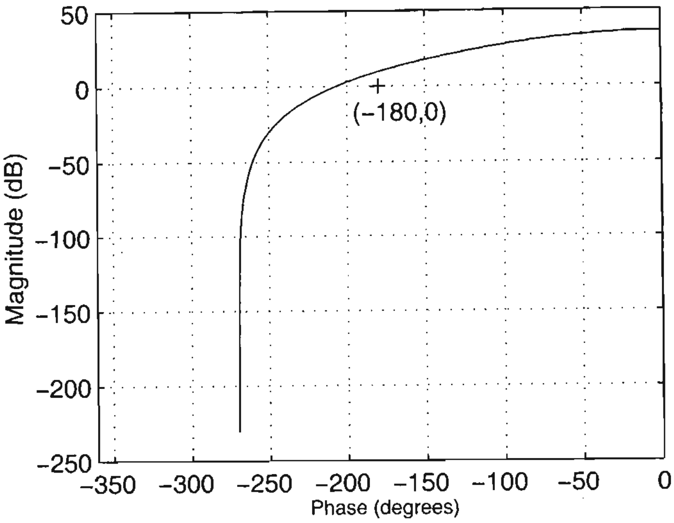
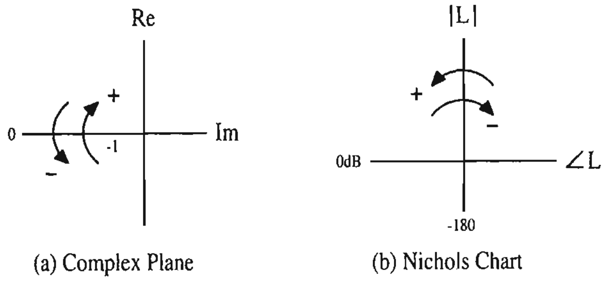


Figure 5.3: The half-sheet Nichols Chart.



(a) Complex Plane

(b) Nichols Chart

Figure 5.4: Stability crossings in the Complex Plane and Nichols Chart.

on the Nyquist diagram occurs if the plot crosses the negative part of the real axis i.e

$$\Re[L(s)] < -1 \tag{5.4}$$

Each crossing has associated with it a sign that indicates the direction of the crossing [9]. Figure 5.4(a) shows the sign convention. The equivalent Nichols Chart crossing is shown in Figure 5.4(b). The Nyquist criterion states that

$$n = Z - N \tag{5.5}$$

where Z = total number of closed-loop poles inside the Nyquist contour, N = number of -1 encirclements in the direction of the contour, and n = total number of open loop poles (including multiplicity) inside the contour. This is translated into the Nichols Chart stability criterion which states that

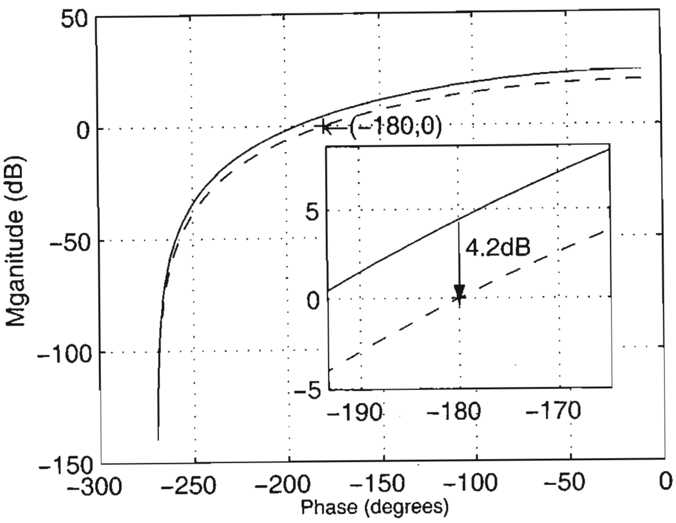


Figure 5.5: Unstable(solid line) and critically stable(dashed line) closed loop plot of $L(s)$.

a feedback system is stable if the plot of the loop transmission $L(s)$ does not intersect the point $(-180^\circ, 0dB)$, and the sum of the crossings of the line

$$Y = \{(\theta, r)\,,\, \theta = -180^\circ\,,\, r > 0dB\} \tag{5.6}$$

is equal to n ($+1$ refers to the line crossing from right to left, and -1 refers to the line crossing from left to right). For the multi-sheeted Nichols Chart, this refers to the sum of the crossings of the line

$$Y + 2k(-180^\circ, 0dB) \tag{5.7}$$

and also $L(s)$ must not intersect $(-180^\circ, 0dB)(2k+1)$ where $k = 1, 2, \ldots$

As an example, consider the following transfer function of the loop transmission [9],

$$L(s) = \frac{k}{(s+1)(s+2)(s+3)} \quad , \; k = 100 \tag{5.8}$$

which is open-loop stable. The Nichols Chart plot is shown in Figure 5.5. It can be seen that the closed-loop system will be unstable for $k = 100$, since $L(s)$ intersects Y. To make $L(s)$ closed-loop stable, the gain must be reduced until $L(s)$ no longer crosses $(-180^\circ, |L(s)| < 0dB)$. To do this, the Nichols Chart must be moved down by $4.2dB$. The dashed line shows the Nichols Chart after it has been moved. The new gain would be $40dB - 4.2dB = 35.8dB$. This new gain occurs at $k = 61.7$. The plant would be critically stable. Thus $L(s)$ is closed-loop stable for $k < 61.7$. The advantage of

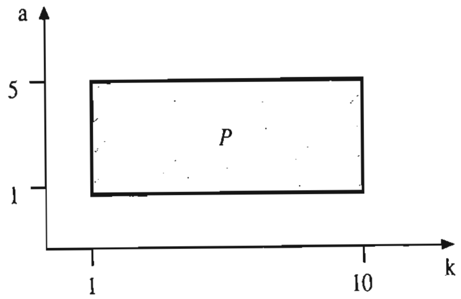


Figure 5.6: Region of plant parameter uncertainty.

using Nichols Chart in QFT is that when plant templates are generated, the templates can be moved without distortion, thereby forming the loop bounds [15]. That is

$$\begin{aligned} |L|_{dB} &= |P|_{dB} + |G|_{dB} \\ \angle L &= \angle P + \angle G \end{aligned} \tag{5.9}$$

The angle of the loop transmission is the critical issue.

5.2 Plant and Loop Transmission Templates

Most plants are subject to parameter variation. Parameter variation may occur via changes in operating conditions, changes in environment, and/or changes in measuring instruments tolerances.

5.2.1 Illustrative example

This example is based on the one given in [9]. Consider the plant described by

$$P(s) = \frac{ka}{s(s+a)} \tag{5.10}$$

where a varies between 1 and 5, and k varies between 1 and 10 i.e.

$$P(s) = \frac{ka}{s(s+a)}\,, \quad k\,,a \in \{(1,10),(1,5)\} \tag{5.11}$$

The region of plant parameter uncertainty is plotted in Figure 5.6. All the plants belong to plant set $\mathcal{P}$. The plant set template $\mathcal{P}$ is mapped to the complex frequency domain, and the template is plotted on the Nichols Chart. This mapping is done by calculating the magnitude and phase of a few plants in the region of parameter uncertainty. The magnitude and phase of these plants are at a single frequency. Table 5.1 illustrates the procedure.

$\omega = 1$			
k	a	Magnitude(dB)	Phase (degrees)
1	1	-3.01	-45.78
1	5	-0.17	-12.09
10	1	16.98	-45.78
10	5	19.82	-12.09

Table 5.1: Magnitude and phase of some selected plants

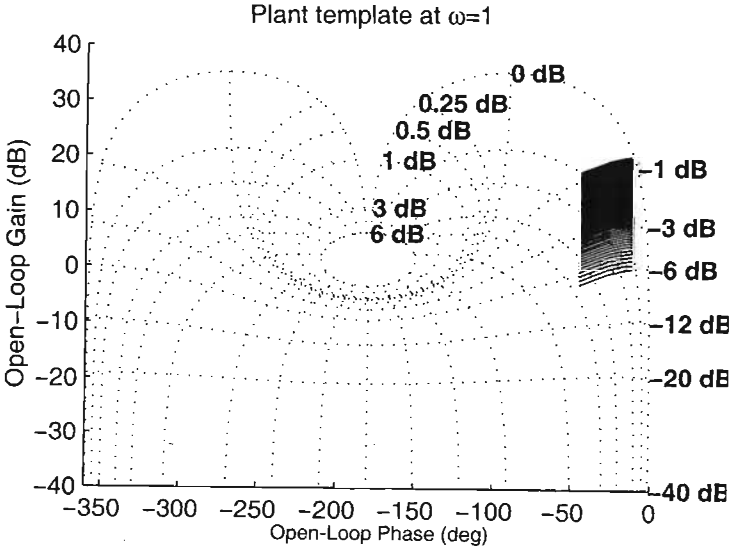


Figure 5.7: Plant template at $\omega = 1$.

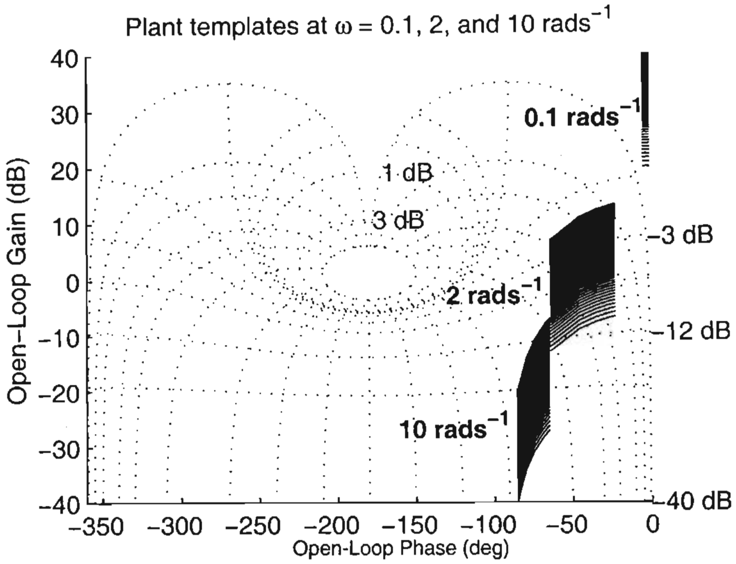


Figure 5.8: Plant templates generated at different frequencies.

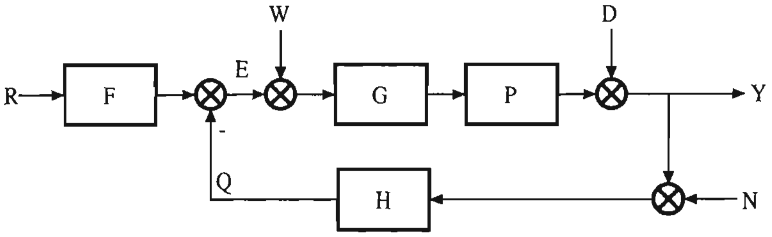


Figure 5.9: A general feedback system.

The magnitude and phase at $\omega = 1$ are plotted on the Nichols Chart. This is shown in Figure 5.7. It is interesting to note that variation in k does not affect the phase of the plant. This fact is exploited when designing controllers. Figure 5.8 shows a Nichols Chart plot of plant templates at various frequencies. It is obvious that the plant template can contain an infinite number of plants. Therefore, a single plant is chosen, and the same plant is used throughout the design process. This plant is usually referred to as the nominal plant.

5.2.2 Loop transmission templates

In the feedback system shown in Figure 5.9, the loop transmission is defined by $L = PG$ where G is a controller with no uncertainty and P is the parametric plant above [9]. Each plant P in $\mathcal{P}$ will give a distinct L. The

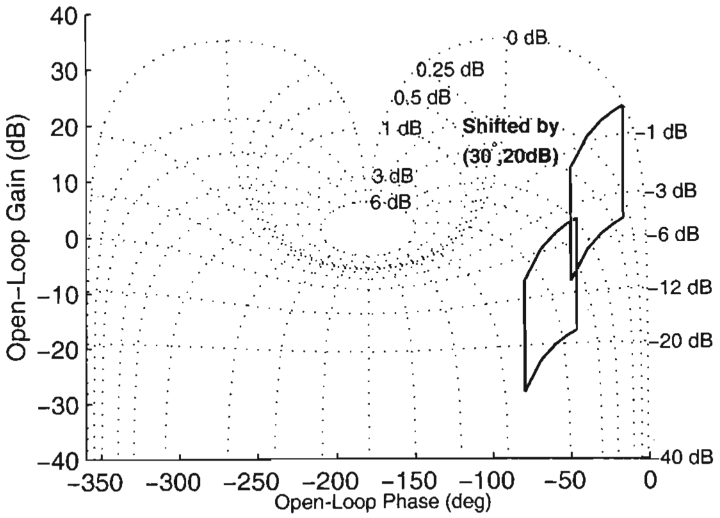


Figure 5.10: The plant template at $\omega = 5$ being shifted by $\{30^\circ, 20dB\}$.

plant template $\mathcal{P}$ therefore gives rise to a loop transmission template $\mathcal{L}(\omega)$. Formally stated,

$$\mathcal{L}(\omega) = \{L = P(\omega)G(\omega)\}\ , \quad P(\omega) \in \mathcal{P} \tag{5.12}$$

For example, at $\omega = 5$, $G(j5) = \{30^\circ, 20dB\}$, then the loop transmission template $\mathcal{L}(j5)$ is obtained by multiplying each complex number in $\mathcal{P}(j5)$ by $G(j5)$. When plotted on the Nichols Chart along with $\mathcal{P}(j5)$, as shown in Figure 5.10, it can be seen that the loop transmission template is offset vertically from the plant template by $20dB$ and horizontally by 30°. There is no distortion of the loop transmission template. Part of the QFT design process is to find suitable locations for $\mathcal{L}(\omega)$ subject to constraints on $\mathcal{P}(\omega)$. This is done by moving $\mathcal{P}(\omega)$ around on the Nichols Chart without invalidating the design constraints. This action generates a bound on $\mathcal{L}(\omega)$.

5.2.3 Bounds

The loci of constant

$$\left|\frac{L}{1+L}\right| \tag{5.13}$$

or closed-loop transmissions on the interior of the Nichols Chart is usually used to generate bounds on the open-loop transmission plot (or, more precisely, on $L(\omega)$). For instance, suppose that $L(\omega)$ is subject to the constraint

$$\left|\frac{L(j5)}{1+L(j5)}\right| \leq 3dB\ \forall\ P\ in\ \mathcal{P} \tag{5.14}$$

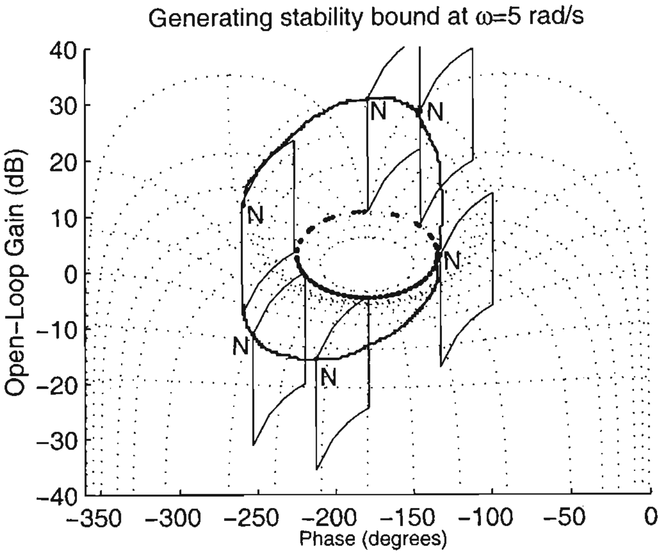


Figure 5.11: Procedure for generating a stability bound.

This means that no part of $L(j5)$ is allowed to lie within the $3dB$ locus of the Nichols Chart. Keeping the vertical sides of the loop transmission template parallel to the vertical axis of the Nichols Chart, the loop transmission template is moved around the M-circle such that its boundary just touches the M-circle. At each point, the nominal plant point on the loop transmission template is marked on the Nichols Chart. These points are then joined to form the $B(j5)$ boundary. Figure 5.11 shows the general idea.

This procedure is carried out for each frequency of interest. This will result in a group of bounds on the Nichols Chart. The design procedure is to now shape the nominal loop transmission transfer function such that it does not lie within these bounds.

5.3 A QFT Example

Consider the single loop feedback system as shown in Figure 5.9. This example is given in [9]. The uncertain plant $P(s)$ is described by the plant set $\mathcal{P}$,

$$\mathcal{P} = \left\{ P(s) = \frac{k}{(s+a)(s+b)} \,, \quad k \in [1, 10] \,, a \in [1, 5] \,, b \in [20, 30] \right\} \quad (5.15)$$

The feedback sensor has unity gain. The specifications of the feedback requirements are as follows:

1. Design a controller $G(s)$ such that the closed-loop system is stable $\forall\, P \in \mathcal{P}$.

2. Ensure that
$$M = \left|\frac{L}{1+L}\right| \leq 3dB \,\forall\, \omega, \ \ \forall\, P \in \mathcal{P} \tag{5.16}$$

3. Satisfy the output disturbance performance criteria
$$\left|\frac{Y(j\omega)}{D(j\omega)}\right| < 0.28 \,\forall\, P \in \mathcal{P}, \ \omega < 0.1 \tag{5.17}$$

The QFT design can be broken down into the following steps:

1. calculate the magnitude and phase of the plant transfer function at the boundaries of the uncertainty region,
2. plot the calculated magnitude and phase values on the Nichols Chart. This will be $P(j1)$,
3. form the stability bounds for $L(j1)$,
4. repeat steps 1 to 3 for every frequency of interest,
5. plot the nominal loop transmission L_n, and
6. shape L_n by adding poles and zeros such that L_n lies as close as possible to the stability bounds.

5.3.1 Magnitude and Phase Calculations

The magnitude of the transfer function is given by

$$\begin{aligned} |P(s)| &= |P(j\omega)| \\ &= \frac{|k|}{|j\omega + a||j\omega + b|} \\ &= 20\log_{10}(k) - 20\log_{10}\sqrt{(\omega^2 + a^2)} - 20\log_{10}\sqrt{(\omega^2 + b^2)} \end{aligned} \tag{5.18}$$

The phase of the transfer function is given by

$$\angle P(j\omega) = \tan^{-1}\left(\frac{0}{k}\right) + \tan^{-1}\left(\frac{-\omega}{a}\right) + \tan^{-1}\left(\frac{-\omega}{b}\right) \tag{5.19}$$

The frequency range of interest is from $\omega = 0.1\,\mathrm{rad\,s^{-1}}$ to $\omega = 30\,\mathrm{rad\,s^{-1}}$. The following frequency vector will be used: $[0.1, 1, 10, 20, 30]$. Table 5.2 shows the magnitude and phase for different plants at each frequency. Figure 5.12 shows the plant templates on the Nichols Chart for each frequency.

$\omega = 0.1$

k	a	b	|P|	∠P
1	1	20	-26.06	-5.99
1	5	30	-43.52	-1.33
10	1	20	-6.06	-5.99
10	5	30	-23.52	-1.33

$\omega = 1$

k	a	b	|P|	∠P
1	1	20	-29.04	-47.86
1	5	30	-43.69	-13.21
10	1	20	-9.04	-47.86
10	5	30	-23.69	-13.21

$\omega = 10$

k	a	b	|P|	∠P
1	1	20	-47.03	-110.85
1	5	30	-50.96	-81.86
10	1	20	-27.03	-110.85
10	5	30	-30.96	-81.86

$\omega = 20$

k	a	b	|P|	∠P
1	1	20	-55.06	-132.13
1	5	30	-57.42	-109.65
10	1	20	-35.06	-132.13
10	5	30	-37.42	-109.65

$\omega = 30$

k	a	b	|P|	∠P
1	1	20	-60.68	-144.40
1	5	30	-62.21	-125.53
10	1	20	-40.68	-144.40
10	5	30	-42.21	-125.53

Table 5.2: Magnitude (dB) and phase (deg.) of different plants at each frequency

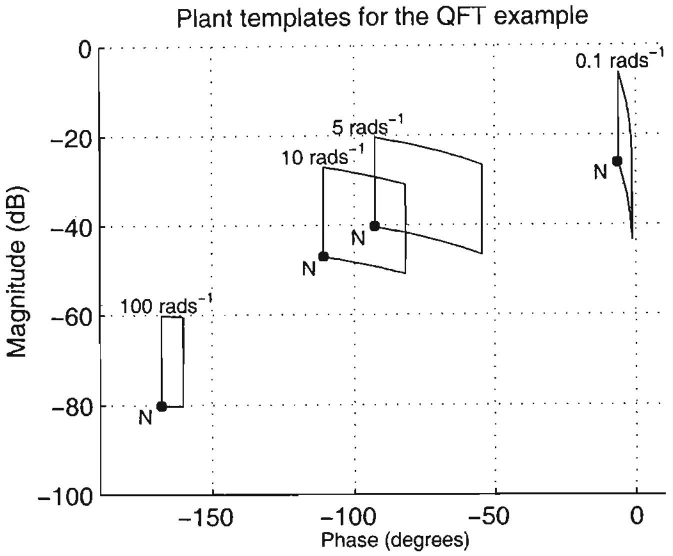


Figure 5.12: The plant templates for the QFT example.

5.3.2 Bounds

Let the nominal plant be

$$P_n = \frac{1}{(s+1)(s+20)} \tag{5.20}$$

The bounds are calculated by moving each plant template around the Nichols Chart to satisfy the control specification. At each point, the nominal plant is marked on the Nichols Chart. This is done by hand. The computer generates the bounds by solving a quadratic inequality at each design phase. Figure 5.13 shows the bounds that are generated.

5.3.3 Loop Shaping

Choose the initial controller to be $G(s) = 1$. Then

$$L_n = P_n G(s) \tag{5.21}$$

The plot of L_n, along with the bounds, is shown in Figure 5.14. L_n needs to satisfy the performance specification for $\omega < 0.1$. This means that the $\omega = 0.1$ point on the loop transmission must lie above the performance bound [9].

The gain of L_n affects its vertical position. By adding a gain of 379, this performance specification can be met. Figure 5.15 shows L_n meeting this specification.

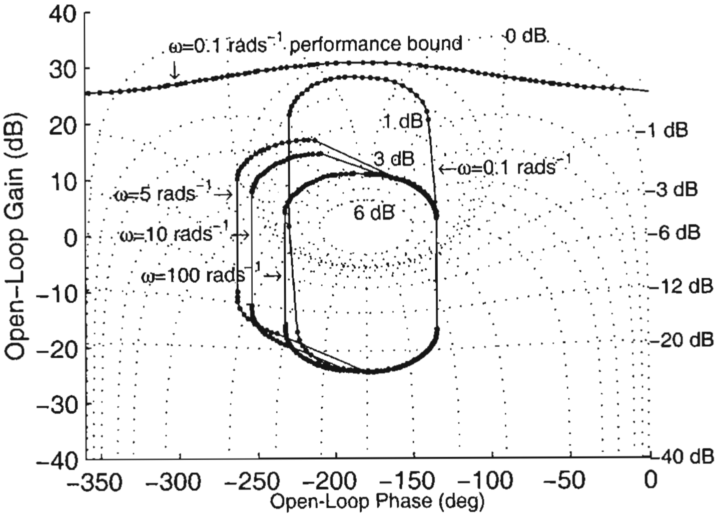


Figure 5.13: The stability and performance bounds that are generated.

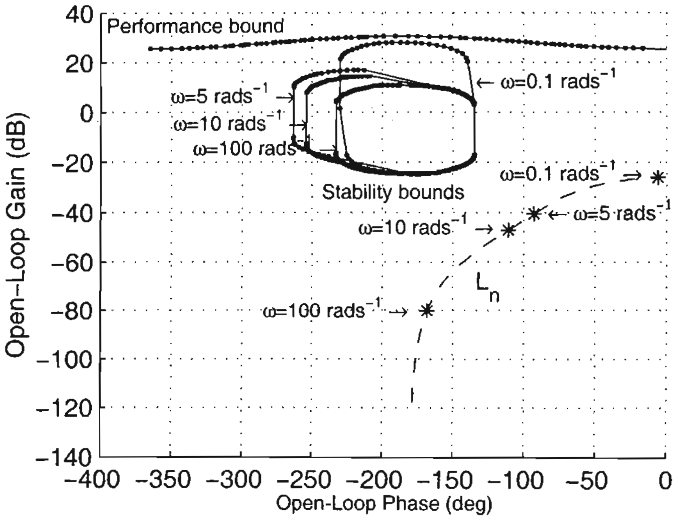


Figure 5.14: Initial plot of L_n.

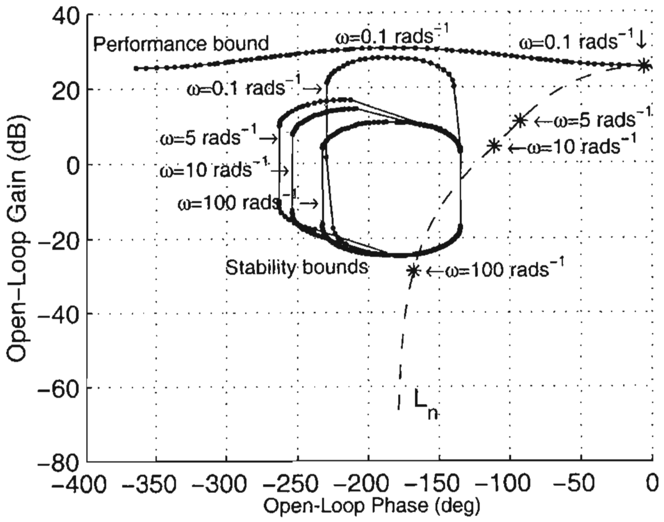


Figure 5.15: L_n satisfying the performance bound.

However, it can be seen that L_n violates the stability boundary. Phase lead is required to move L_n out of the boundary. This means that a zero must be added to L_n. The design requires $45° + 5°$ reserve at $\omega = 50\,\mathrm{rad\,s^{-1}}$

Add a zero with a value of 42. This results in 50° phase lead. The reason for such a large phase lead is that a pole needs to be added later to make the controller at least proper. Figure 5.16 shows the result of adding a zero. L_n no longer violates the stability bounds. To finish the design, it is preferable to have the high frequency magnitude response of L_n to be as close to the stability boundary as possible, and to drop as quickly as possible. A first order pole can be used for this. The design requires about 40° phase lag at $\omega = 140\,\mathrm{rad\,s^{-1}}$. Choosing a value of 165 for this pole gives the required phase lag. L_n now becomes

$$L_n = \frac{379(\frac{s}{42} + 1)}{(\frac{s}{165} + 1)(s + 1)(s + 20)} \tag{5.22}$$

The controller $G(s)$ is given by

$$G(s) = \frac{L_n}{P_n} \tag{5.23}$$

Figure 5.17 shows the final design.

5.4 The QFT Toolbox

The QFT design process can be long and laborious, especially if a lot of iterations have to be done. Fortunately, there are software tools available

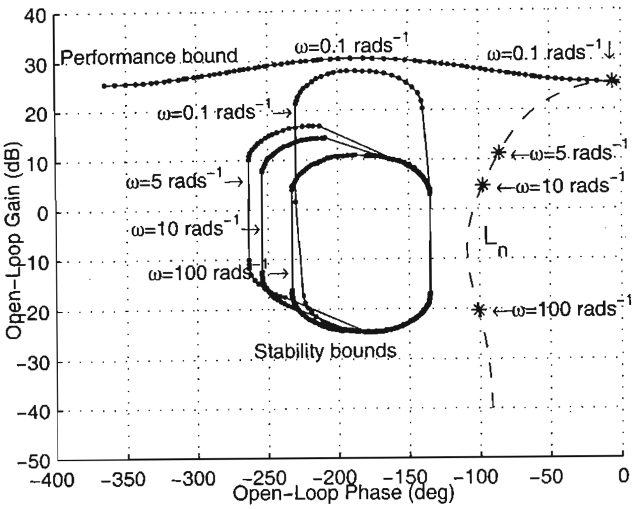


Figure 5.16: The effect of adding a zero to L_n.

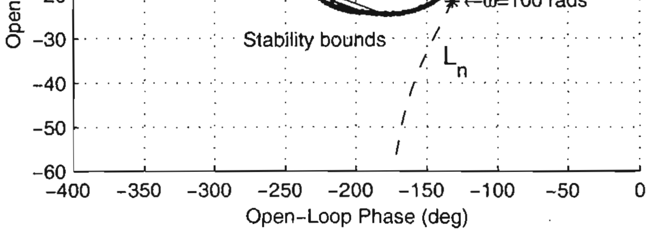


Figure 5.17: The final QFT design.

that ease the design process. One of these is the QFT Toolbox for Matlab [9]. To illustrate the ease and transparency of doing a QFT design by software, the previous example will be used. First, create a numerator matrix that holds 40 values of k between 1 and 10. Then, create a denominator matrix that holds 40 coefficient values of the transfer function denominator, where a and b are between the parametric uncertainty.

Select a matrix index to point to a nominal plant e.g.

```
>>np=21
```

Define the frequency vector

```
>>w=[0.1,1, 10, 20, 30]
```

Calculate the frequency response of the entire plant set by calling

```
>>P=freqcp(num,den,w)
```

The plant templates can then be viewed by typing

```
>>plottmpl(w,w,P,np)
```

The function for generating stability bounds is called as follows

```
>>bdb1=sisobnds(1,w,w,1.2,P,0,np)
```

To group the bounds together, the command

```
>>bdb=grpbnds(bdb1)
```

is executed. The resulting bounds can be viewed with

```
>>plotbnds(bdb)
```

The worst case bound can be found by calling

```
>>ubdb=sectbnds(bdb)
```

Lastly, the interactive loop shaping environment is invoked by calling

```
>>lpshape(w,ubdb,nPn,dPn,del0,nCn,dCn,phs)
```

Refer to the QFT Toolbox manual [9] for information on the parameters to be passed to these QFT commands. Figures 5.18, 5.19, and 5.20 show the three steps of generating plant templates, bounds, and shaping L_n. The numbers in the diagrams refer to the frequencies in radians per second.

From now on, all QFT designs will be carried out using the QFT Toolbox because it is faster and easier.

5.5 Summary

The design of controllers via QFT offers some advantages over other control design methods. The designer can see the trade-offs between specifications and controller design. Since QFT uses uncertainty as one of the design criteria, variations in the parameters of the model describing the system

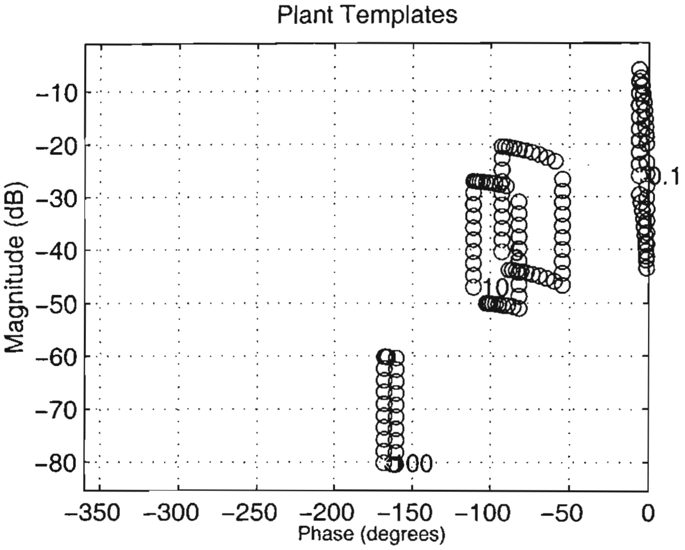


Figure 5.18: Plant templates as generated by the QFT Toolbox.

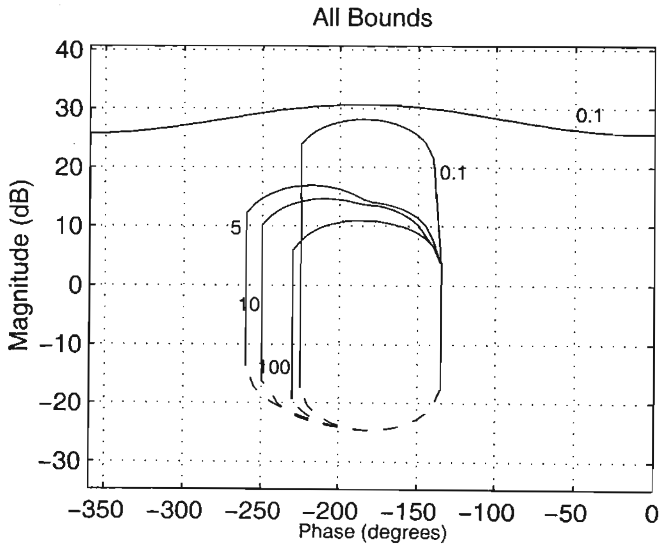


Figure 5.19: The stability and performance bounds.

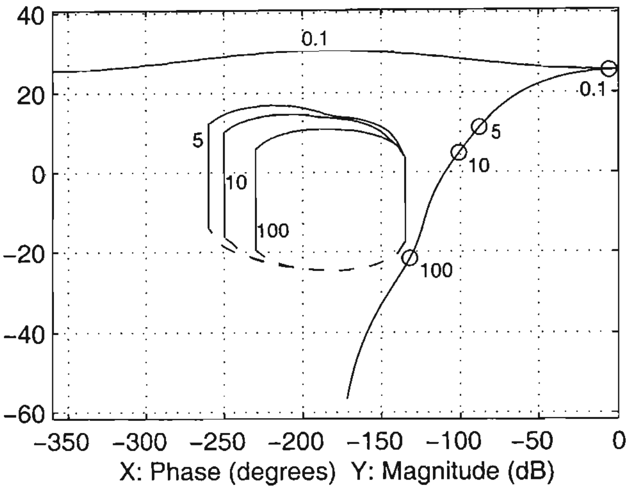


Figure 5.20: Final design using the interactive loop shaping environment.

will result in a design that is robust stable. QFT design operates in the frequency domain, and as such, all that is needed to do a successful design is the frequency response of the plant. Thus plants whose frequency responses are known can be controlled without developing an analytical model. In some cases, it may be impossible to find an analytical model, and this is where QFT is most powerful. The use of QFT CAD packages is helpful, but an essential understanding of how QFT works is necessary to produce successful designs. Controllers obtained from the design process are often simple and can be synthesised in either analog or digital form. The simplicity and low cost of the designed controllers makes QFT a suitable design method for industrial control and processes.

Chapter 6

The Laboratory Power System

> God not only plays dice. He also sometimes throws the dice where they cannot be seen.
>
> STEPHEN WILLIAMS HAWKING
> *Nature*

In order to test the feasibility of using QFT to design robust power system stabilisers, a practical setup will be used.

The setup will consist of an analog controller which will be used in a micro-machine power system laboratory. The micro-machine power system laboratory is situated on the third floor in the School of Electrical and Electronic Engineering at the University of Natal. The laboratory was developed and built by Professor R.G Harley [1], and Professor D.J Limbeer [28], [27]. Various researchers have since then used the laboratory for their work [3], [4], [13], [18], [22]. Micro-Alternator 2 is used in this thesis. It is on the left hand side when facing the control panel. The power system laboratory consists of a synchronous generator, a turbine simulator, which is an analog electronic circuit, a transmission line simulator, and an infinite bus, which in essence, is a 3-phase voltage supply. Associated with the power system setup are the control circuits which are the governor and automatic voltage regulator.

It is only necessary to obtain a set of models describing the behaviour of these components so that they can simulated in conjunction with the synchronous machine model developed in Chapter 2. Thus the expected response of the physical system to various disturbances can be visualised before the actual experiment is carried out. A computer simulation also allows for the QFT controller to be used with the power system model, thereby giving the designer useful insight into potential problems and trade-offs. The idea is to relate the theory of power system stabilisers to computer simulation. The computer simulation can be used to relate the simulation model to the practical power system. The circle is closed when the practical system is used to verify the theory of power system stabilisers. Thus one would be able to go back and forth between theory and simulation, simulation and practice, and practice and theory.

This thesis only describes the transfer function models of the existing power system components in the laboratory. It does not offer explanations as to why the components were designed with particular values, nor does it explain the construction of such circuits. Details on these topics can be found in [27], [37], and [33].

The following power system components will be discussed:

1. the micro-alternator (synchronous generator),
2. Time Constant Regulator (TCR),
3. AVR and Governor,
4. turbine simulator,
5. the distributed mass shaft,
6. transmission line simulator,
7. the transducers, and
8. data capture equipment.

6.1 The micro-alternator

The micro-alternator is a Mawsley 3kW salient 4-pole synchronous generator. It is driven by a dc motor whose characteristics are altered by the turbine simulator. Various research efforts have been undertaken to determine the parameters of the micro-alternator [22]. The results are summarised in Table 6.1. These parameters can be used in the synchronous machine model from Chapter 2 to simulate the micro-alternator.

The parameters are given in per unit, and no conversion to SI units is necessary, as the PSB synchronous machine model accepts per unit values. The per unit system that is used can be found in Appendix A. The per unit synchronous generator Simulink model must be chosen when using the Power System Blockset. If SI units are required, then they can be found in [43]. The micro-alternator is star-connected, and is rated 220V line to line.

Micro Alternator Parameters		
Parameter	Value (p.u)	Value (Ω)
X_l	0.11	1.78
X_{md}	1.98	31.97
X_{mq}	1.87	30.19
X_d	2.09	33.75
X_q	1.98	31.97
R_s	0.006	0.097
X_f	0.1	1.61
R_{fd}	0.00099	0.016
X_{kd}	0.125	2.02
R_{kd}	0.0212	0.34
X_{kq}	0.257	4.15
R_{kq}	0.027	0.44

Table 6.1: A summary of the micro alternator parameters [22].

6.2 Time constant regulator

The micro-machine has to represent an actual synchronous generator in terms of its electrical parameters. This allows for experimentation and tests to be carried out on the micro-machine, rather than, say, a 300MW synchronous generator in a power generation plant. The micro-machine actually models a 1000MW generator situated in Koeberg, South Africa [27]. The electrical parameters are in per unit for the micro-machine, and for the actual machine.

However, when scaling of the per unit values are done, the field resistance of the micro-machine is too high compared to the rest of the per unit values i.e the time constant for the field circuit is too high. Hence its electrical characteristics are different from the actual machine. In order to reduce the field winding resistance, the technique of introducing "negative resistance"is used.

This "negative resistance" or more accurately, lower resistance, is brought about by altering the inductance of the field winding, which in turn affects the winding's time constant. A reduction in the time constant results in an "apparent" decrease in field resistance, all other factors being constant. The variation in inductance is achieved by winding a shadow coil in close proximity to the field winding. The magnetic field of the shadow winding is varied by a chopper circuit.

The magnetic flux linkage between the field winding and the armature winding is nonlinear. A time constant setting that gives a time constant of τ at a lightly loaded operating point will give a different τ for a heavily loaded operating point at the same time constant setting.

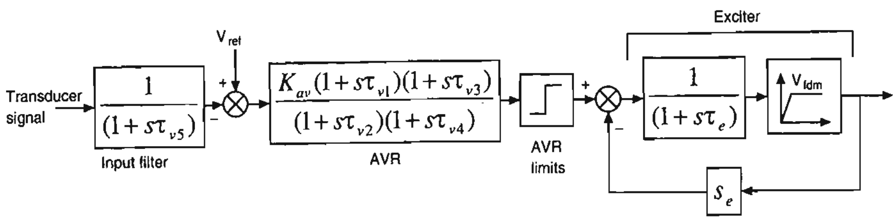

Transducer signal
V_{ref}
+
-
$\frac{1}{(1+s\tau_{v5})}$
Input filter
$\frac{K_{av}(1+s\tau_{v1})(1+s\tau_{v3})}{(1+s\tau_{v2})(1+s\tau_{v4})}$
AVR
AVR limits
+
-
Exciter
$\frac{1}{(1+s\tau_e)}$
V_{fdm}
S_e

Figure 6.1: Block diagram representation of the laboratory AVR.

Laboratory AVR parameters		
Parameter	Value	
τ_{v1}	0.616	sec
τ_{v2}	2.178	sec
τ_{v3}	0.199	sec
τ_{v4}	0.039	sec
τ_{v5}	0.0235	sec
τ_e	0.49	sec
K_a	206	

Table 6.2: The parameter values for the different components of the AVR.

Various measurements were taken of the terminal voltage rise and fall times to a step change in the AVR reference voltage in an attempt to calibrate the TCR. The measurements were done at different TCR settings for a low operating point and a high operating point. The results are shown in Table 6.10. Details on the construction of the TCR and the theory behind it can be found in [33].

6.3 AVR and Governor

The laboratory power system governor is part of the turbine simulator, and will be described in the next section. The AVR used in the laboratory is an electronic circuit that generates and controls the field voltage. The AVR consists of an input filter to filter the output of the terminal voltage transducer, a compensator with hard limits to provide regulation, and an exciter amplifier with an upper limit on the field output voltage. Figure 6.1 shows the block diagram representation of the AVR, along with the relevant transfer function for each component. It is only necessary to use the compensator block with the exciter for generating the AVR model, since the input filter has a much shorter time constant and does not influence the AVR dynamics as much as the other transfer function blocks. The values of the time constants are shown in Table 6.2.

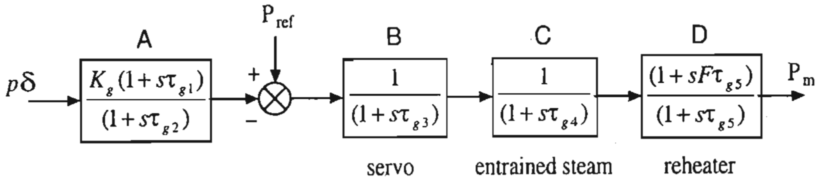

P_{ref}
A
B
C
D
$p\delta$
$\frac{K_g(1+s\tau_{g1})}{(1+s\tau_{g2})}$
+
-
$\frac{1}{(1+s\tau_{g3})}$
$\frac{1}{(1+s\tau_{g4})}$
$\frac{(1+sF\tau_{g5})}{(1+s\tau_{g5})}$
P_m
servo
entrained steam
reheater

Figure 6.2: Block diagram of the steam turbine simulator.

6.4 Turbine simulator

To simulate a steam turbine as a prime mover in the power system laboratory, a DC motor with power control is used. The DC motor is coupled to the generator via the distributed mass shaft, and is driven by a steam turbine simulator circuit.

The reference to the steam turbine simulator circuit comes from a phase advance compensation circuit. The DC motor speed is monitored by a tachometer, and the speed signal is fed to a summation amplifier in the phase advance compensation circuit. This circuit can be found in [27]. Therefore, under closed loop control, the DC motor characteristics are made to represent a steam turbine characteristics. Figure 6.2 shows the block diagram representation of the turbine simulator. The turbine characteristics that are simulated are:

1. the phase advance compensation. This is denoted by (A) in Figure 6.2. This accounts for the phase lag between the turbine speed and required power output in a real turbine,

2. the steam valve. This is denoted by (B) in Figure 6.2. This valve is controlled by a servo in the actual steam turbine,

3. entrained steam delay. This is denoted by (C) in Figure 6.2. This is the time associated with moving steam from the intake of the steam turbine to the turbine reheat section, and

4. turbine reheater. This is denoted by (D) in Figure 6.2. It adds superheat. This component usually has a thermal time constant associated with it.

All these characteristics can be modelled by transfer functions. Saturation is not modelled. These transfer functions are synthesised to form the analog control circuit. The transfer function form in the block diagram representation is ideal for computer simulation, since the block diagram can be easily created in Simulink. The values for the turbine simulator are shown in Table 6.3, along with a description of what they represent.

Turbine Simulator Values			
Constant	Value		Description
τ_{g1}	0.264	sec	phase advance compensation
τ_{g2}	0.0264	sec	phase advance compensation
τ_{g3}	0.15	sec	servo time constant
τ_{g4}	0.594	sec	entrained steam delay
τ_{g5}	2.662	sec	steam reheat constant
F	0.322	p.u	shaft output ahead of reheater

Table 6.3: The values and descriptions of the various turbine components.

6.5 Distributed mass shaft

The distributed mass shaft couples the DC motor to the micro-alternator. It was originally designed to simulate the transmission shaft of a particular generator in the Koeberg power generation plant [27]. It was initially and subsequently used in studies of subsynchronous resonance [35], and is now a permanent component of the power system setup in the laboratory.

The shaft consists of a 15mm diameter steel shaft, which runs through, and is attached to, five metal disks. Each of these disks have a specific moment of inertia as shown in Table 6.4. Thus the entire distributed mass shaft can be described by a matrix differential equation (Equation 6.1) that relate the torques to the moments of inertia, and damping and viscous coefficients.

$$0 = \boldsymbol{J}\ddot{\boldsymbol{\delta}} + \boldsymbol{D}\dot{\boldsymbol{\delta}} + \boldsymbol{K}\boldsymbol{\delta} + \boldsymbol{T} \tag{6.1}$$

Rearranging,

$$-\boldsymbol{J}\ddot{\boldsymbol{\delta}} = \boldsymbol{D}\dot{\boldsymbol{\delta}} + \boldsymbol{K}\boldsymbol{\delta} + \boldsymbol{T} \tag{6.2}$$

For the 6-state laboratory shaft system,

$$\boldsymbol{J} = \begin{bmatrix} J_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_6 \end{bmatrix} \tag{6.3}$$

$$\boldsymbol{D} = \begin{bmatrix} D_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & D_6 \end{bmatrix} \tag{6.4}$$

$$\boldsymbol{K} = \begin{bmatrix} K_1 & -K_1 & 0 & 0 & 0 & 0 \\ -K_1 & K_1 + K_2 & -K_2 & 0 & 0 & 0 \\ 0 & K_2 & K_2 + K_3 & -K_3 & 0 & 0 \\ 0 & 0 & K_3 & K_3 + K_4 & -K_4 & 0 \\ 0 & 0 & 0 & K_4 & K_4 + K_5 & -K_5 \\ 0 & 0 & 0 & 0 & -K_5 & K_5 \end{bmatrix} \tag{6.5}$$

Distributed Mass Shaft			
	Inertia	Damping Coeff.	Viscous Coeff.
Disk 1	2.355e-3 (J_1)	7.4e-4 (D_1)	
Disk 2	7.608e-3 (J_2)	2.39e-3 (D_2)	10.63 (K_1)
Disk 3	7.576e-3 (J_3)	2.38e-3 (D_3)	25.34 (K_2)
Disk 4	7.779e-3 (J_4)	2.44e-3 (D_4)	23.42 (K_3)
Disk 5	1.063e-2 (J_5)	3.34e-3 (D_5)	26.93 (K_4)
Shaft	2.192e-5 (J_6)	6.89e-5 (D_6)	70.53 (K_5)

Table 6.4: Data for the distributed mass shaft in p.u.

$$\ddot{\boldsymbol{\delta}} = \begin{bmatrix} \ddot{\delta}_1 \\ \vdots \\ \ddot{\delta}_6 \end{bmatrix}, \dot{\boldsymbol{\delta}} = \begin{bmatrix} \dot{\delta}_1 \\ \vdots \\ \dot{\delta}_6 \end{bmatrix}, \boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_6 \end{bmatrix} \tag{6.6}$$

$$\boldsymbol{T} = \begin{bmatrix} T_m & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{6.7}$$

In Equation 6.1, $\boldsymbol{J}$ and $\boldsymbol{D}$ are diagonal matrices. The shaft can present a potential problem in power system stabiliser design. Its dynamics add to the complexity of the overall power system model. The distributed mass shaft may introduce considerable phase lag between the prime mover mechanical power input and the applied mechanical power to the generator shaft. This may invalidate the power system stabiliser design which uses a power, or speed, signal as feedback from the prime mover. The program that simulates the shaft dynamics is given in Appendix F.16.

The problem may be alleviated if the speed measurement at the generator, or the phase lag of the distributed mass shaft is measured. The designed power system stabiliser can then take into account this extra phase lag. Table 6.4 gives the list of constants for the disks and shaft.

6.6 Transmission line simulator

The transmission line simulator in the laboratory can simulate a transmission line up to a length of 300km. Different values of inductances and resistances can be selected. The transmission line simulator consists of a set of 220V reactors, each one with a known inductance and resistance. There is a set of reactors for each phase. Thus a transmission line may be simulated with different values for each phase.

The inductance and resistance values for a reactor set is shown in Table 6.5. There are 4 reactors making up a set. Thus there are $2^4 = 16$ different values of inductances and resistances to chose from. The reactors in a set are

Transmission Line Simulator		
Reactor Set	Reactance (Ω)	Resistance (Ω)
1	1.09	0.076
2	2.04	0.115
3	4.02	0.155
4	8.75	0.206

Table 6.5: The four reactors and their respective values.

connected in series. On the laboratory control panel, the values are made up by connecting the different reactors together with cable connectors.

6.7 Transducers

The laboratory has various transducers for measuring voltage, current, load angle, and speed.

The transducers that will be considered are the ones that measure terminal voltage, speed deviation, and load angle. The terminal voltage transducer is a full wave rectifier connected to a step down transformer. The rectified AC voltage is converted to smoothed DC voltage via a filter circuit. The output of the terminal voltage transducer is therefore in (volts dc)/(volts ac).

The speed deviation transducer is a digital optical circuit that measures the change in shaft speed of the generator with a shaft encoder. The output of the speed deviation transducer is therefore in (volts dc)/(rpm).

The load angle transducer is an analog circuit that measures the difference in phase voltages from a two-phase tachometer connected to the alternator shaft. One phase is the reference phase, and it is calibrated by aligning the reference coil with the A-phase coil of the alternator. The output of the load angle transducer is therefore in (volts dc)/(electrical degree).

The load angle can also be measured visually while the alternator is rotating. A stroboscope is used to create a stationary image of the rotating angle gauge affixed to the shaft end. The load angle is the difference between the reference electrical angle and the rotor angle.

6.8 Data capture equipment

The data capture equipment consists of an Eagle Technology PC30 data capture card [42].

The data capture card plugs into the PC. It has a 12-bit, 16 channel, analog to digital converter. The input voltage range for each channel is −5Vdc to +5Vdc. Therefore all voltage outputs from the transducers must be scaled to lie within this voltage range. This voltage scaling is done with a gain box.

The gain box is a shielded metal box with all the input BNC connectors on one side and all the BNC output connectors on the opposite side. Inside the gain box is an operational amplifier circuit for each input-output pair. The op-amp circuit have different fixed resistors for different gain settings. The resistor for a specific gain value is selected by setting a DIP switch.

The data capture software for the PC allows simultaneous capture of all input channels at a specific sampling frequency. There are also other configurations of the sampling rate and number of channels. The captured data file can be saved as a text file, or a Matlab `.mat` file.

6.9 Calibration

It is essential that prior to performing an experiment in the power system laboratory, the input/output relationship for a transducer be quantified. This relationship is measured and verified by running the alternator at different operating points and calibrating the relevant transducers. The relationship between the speed and speed deviation transducer is shown. The relationship between the AVR voltage reference and the terminal voltage is shown. The relationship between the terminal voltage and the terminal voltage transducer is shown, as well as the load angle and load angle transducer relationship.

6.9.1 Speed deviation

The alternator was run unsynchronised. The rotor speed was varied from 1450 rpm to 1550 rpm, in increments of 10 rpm. Synchronous speed is 1500 rpm. Table 6.6 shows the speed deviation transducer voltage output for each speed setting. Figure 6.3 shows the relationship in a graphical form.

6.9.2 Load angle

The alternator was synchronised to the infinite bus. The load angle was varied by changing the active power transfer. The excitation voltage was also changed to ensure that the alternator did not become under-excited. The results are tabulated in Table 6.7, and plotted in Figure 6.4. It can be seen from the resulting graphs that the speed transducer and the load angle transducer are linear.

Calibration of Speed Transducer	
Speed (rpm)	Transducer output
1450	-900mV
1460	-709mV
1470	-559mV
1480	-393mV
1490	-212mV
1500	-50mV
1510	112.5mV
1520	259.4mV
1530	421.9mV
1540	606.2mV
1550	790.6mV

Table 6.6: Measured data from the speed deviation calibration.

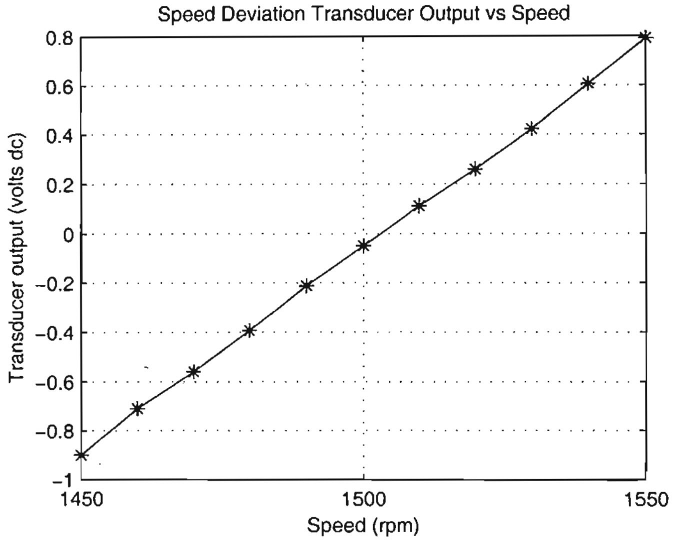


Figure 6.3: Graphical representation of the speed vs. speed deviation .

Calibration of Load Angle Transducer	
Load Angle (deg)	Transducer output
0	0V
5	0.234V
10	0.512V
15	0.759V
20	0.962V
25	1.219V
30	1.441V
35	1.688V
40	1.944V
45	2.184V
50	2.425V
55	2.638V
60	2.875V

Table 6.7: The load angle transducer calibration data.

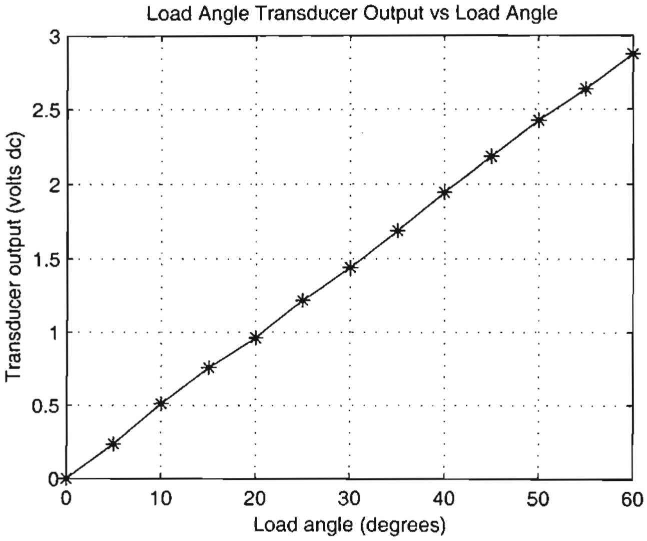


Figure 6.4: Relationship between load angle and load angle transducer.

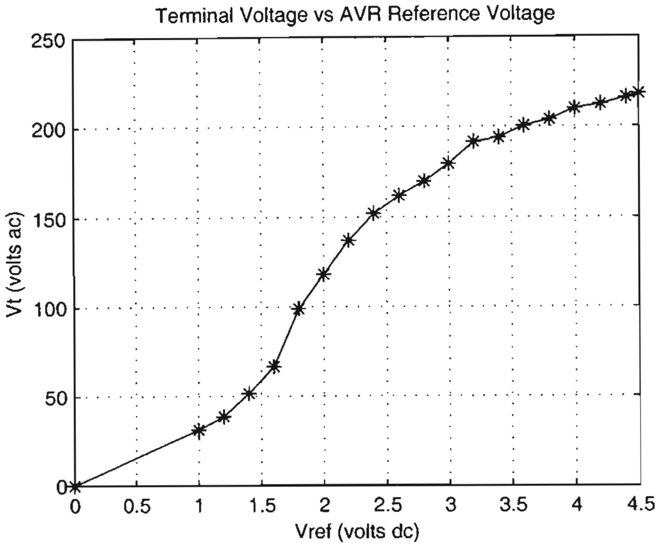


Figure 6.5: The relationship between the AVR ref. voltage and the terminal voltage.

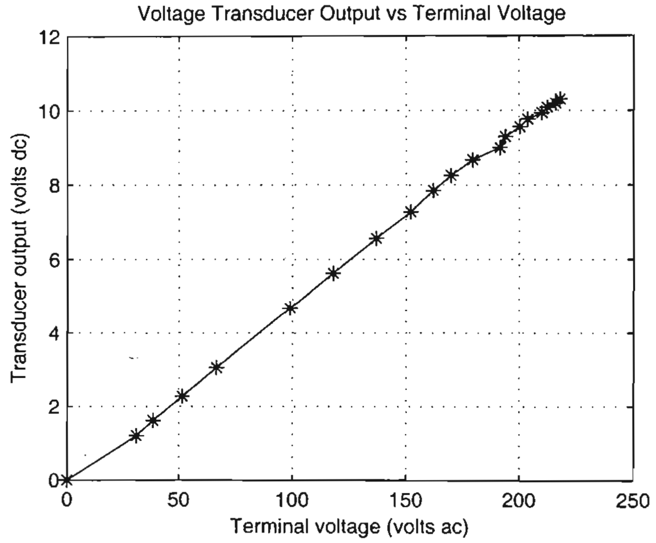


Figure 6.6: The relationship between terminal voltage and terminal voltage transducer output.

Short Circuit Test	
Field Current (amps)	Armature Current (amps)
1.85	4.4
1.7	7.44
1.5	6.4
1.4	6.08
1.25	5.6
1.05	4.96
1	4.32
0.85	3.36
0.7	3.12
0.5	2.4

Table 6.8: Data from the short circuit test.

6.10 Machine tests

Short circuit test, saturation curve measurement, and open circuit time constant measurements were performed on the alternator. The aim was to obtain data for simulation models and simulation environments that would accurately represent the equipment in the power system laboratory.

6.10.1 Short circuit test

This is a standard machine test. The alternator terminals were short-circuit to ground. The armature current was varied from 2.4 amps to rated full load current of 7 amps by changing the field current. The armature current was measured using the analog ammeter with a scale factor of 4. The results are tabulated in Table 6.8. The graph is shown in Figure 6.7.

6.10.2 Saturation curve

This is also a standard machine test. The machine open-circuit saturation curve was determined by varying the DC input to the AVR. This varied the terminal voltage and the field current was monitored. During each measurement, the speed had to be kept at synchronous speed.

The AVR reference voltage was varied from 1.4V dc to 4.6V dc in steps of 0.2V dc. From the AVR-terminal voltage calibration in the previous section, this gave a terminal voltage range of 51V ac to 220V ac. Table 6.9 gives the terminal voltage and field current measurement, along with the AVR reference voltage at which the measurement was done. A plot of field current vs. terminal voltage is shown in Figure 6.8.

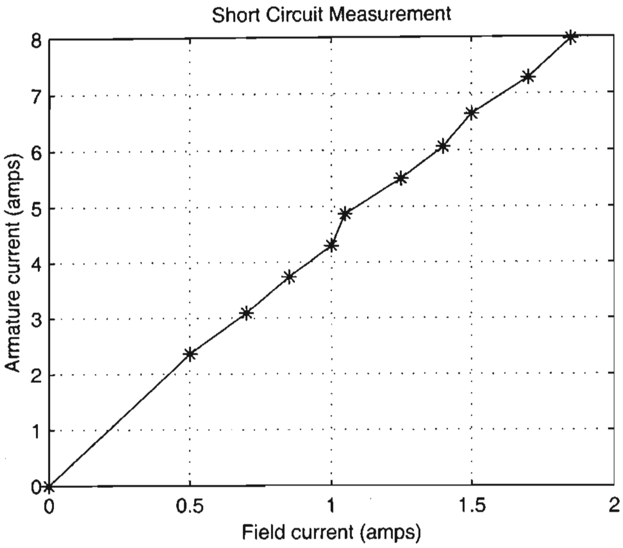


Figure 6.7: Graphical representation of the short circuit results.

Saturation Curve Data		
AVR ref. (vdc)	Field Current (amps)	Terminal Voltage (vac)
1.4	0.40	51
1.6	0.50	66
1.8	0.70	99
2.0	0.80	118
2.2	1.00	137
2.4	1.14	152
2.6	1.30	162
2.8	1.46	170
3.0	1.62	179
3.2	1.80	191
3.4	2.00	194
3.6	2.18	200
3.8	2.38	204
4.0	2.50	210
4.2	2.70	212
4.4	2.90	216
4.6	3.02	218

Table 6.9: Results of the saturation curve measurement.

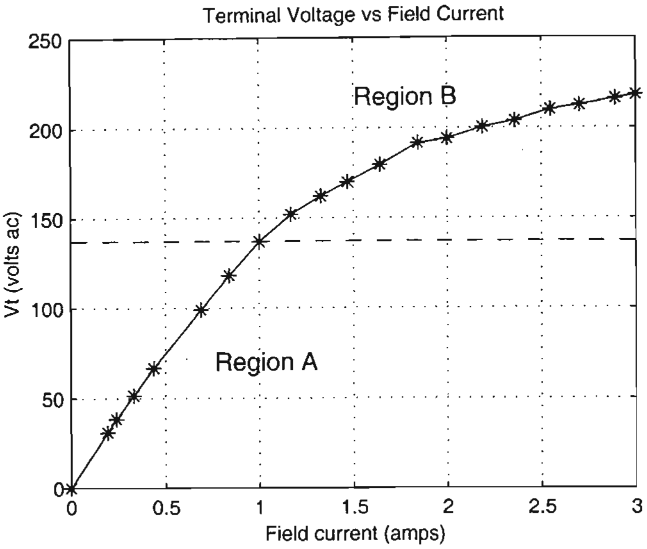


Figure 6.8: The saturation curve of the laboratory alternator.

6.10.3 Open-circuit time constant

The region of the saturation curve in which the alternator (or any synchronous generator) operates affects the open circuit voltage time constant of the machine. A set of tests were done to determine the open-circuit voltage time constant for a particular TCR setting at the low end of the saturation curve (denoted region A in Figure 6.8) and the high end of the saturation curve (denoted region B in Figure 6.8). Because of hysteresis, the time constant is measured for both a step-up and a step-down change in AVR input for each region. A Hewlett Packard digital oscilloscope is used as the measuring instrument. The procedure developed for time constant measurement is as follows.

A step voltage is applied to the AVR input via a step box. The step box is a circuit that generates a step between a lower and upper voltage when a switch is thrown. The magnitude of the step is taken to be 5% of the terminal voltage.

The results of the experiment are outlined in Table 6.10 for selected values of the TCR setting. It can be seen that the time constant for a specific TCR setting varies drastically between the two regions (80V ac and 200V ac). A graph showing the unsaturated and saturated time constants versus different TCR settings is illustrated in Figure 6.9 for step up and step down respectively.

A control designer has to be aware of this phenomenon, and has to design the controller carefully and only for one operating point when using con-

	Open-Circuit Time Constant		
TCR Position	Region	Step up τ (sec)	Step down τ (sec)
1	A	4.24	4.24
	B	1.24	1.72
2	A	3.96	4
	B	1.36	1.52
3	A	3.56	3.72
	B	1.24	1.28
4	A	3.4	3.48
	B	1.08	1.32
5	A	3.64	3.2
	B	1.08	1.2
6	A	3.24	3.28
	B	0.88	1.2

Table 6.10: The time constants measured at different TCR settings.

ventional control design methods. What happens if the operating point for which a controller must be designed lies on the boundary of the two regions?

QFT eliminates these problems and allows the designer to develop a controller that will work in both regions. The reason for this is that the difference in time constants is absorbed into the uncertainty region that is inherent in QFT. But if there is hysteresis, there is a nonlinearity. Describing functions analysis may be of some help in analysing the behaviour.

6.11 PSB Model vs. Physical Model

In order to verify the PSB simulation model, the experiment described in [27] was simulated in Simulink. Figure 6.10 shows the laboratory setup used in [27]. In the experiment, X_c, R_{12}, X_{12}, and X_v were not used. The fault was applied at point A in Figure 6.10.

The simulation results were compared against the experimental results obtained in [27]. The simulation results were also verified in the laboratory by setting up the power system and performing the short-circuit test as described in [27]. The data from the practical, along with the simulation results, are shown in Figure 6.11. The nonlinear generator and transmission line equations can be simulated with the calculated derivatives sent

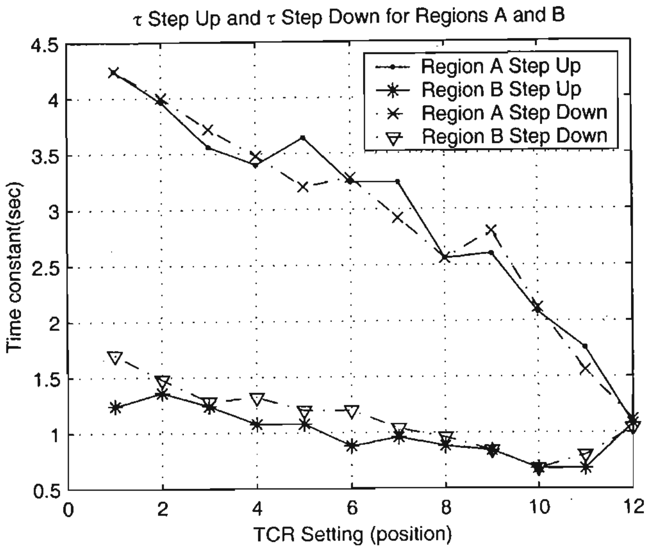


Figure 6.9: Time constants measured during step up and step down.

through an integrator and the state variables fed into the input of the m-file block. The program for this m-file block can be found in Appendix F.6. Figure 6.12 shows the Simulink setup. Since these equations are used in the PSB, there should be no difference in simulation output. A simulation was created to verify this. The PSB model and the m-file model ran in the same simulation environment, and the outputs of each model was sent to the workspace for comparison. Figures 6.13 and 6.14 shows the simulation model and simulation results respectively.

6.12 Summary

The different components available to build a power system in the laboratory have been outlined. Measurements were done to determine the behaviour and characteristics of the transducers. The measured values were used to generate linear equations for each quantity (load angle, speed, etc), using Matlab [29] curve-fitting functions. These equations are used in the simulation to model various components of the power system laboratory. Alternatively, the measured values can be used in a look-up table block in Simulink when creating a Simulink model to represent the power system. Care must be taken to ensure that the simulation model does not request a value that lies outside the measured value range. This method is not used in this thesis. Tests were carried out to determine machine behaviour and characteristics. The tests showed how parameter variations and operating points affected machine performance.

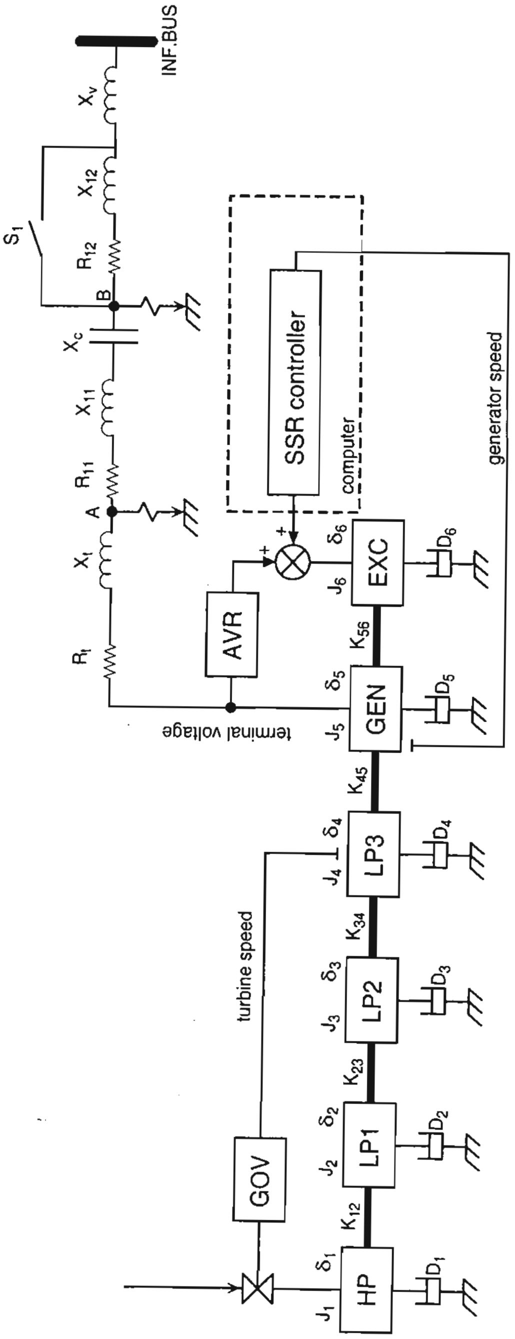


Figure 6.10: The laboratory test system used in [27].

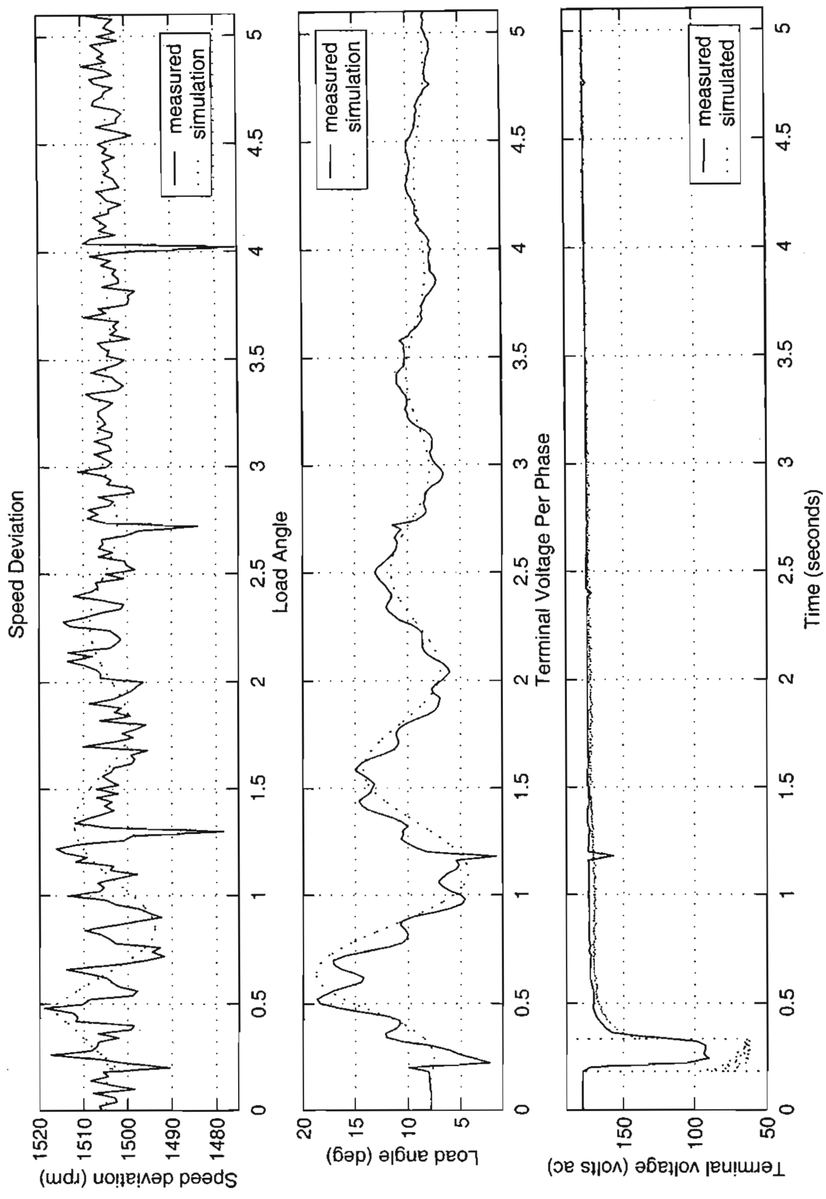


Figure 6.11: Verification of the simulation model against the laboratory power system.

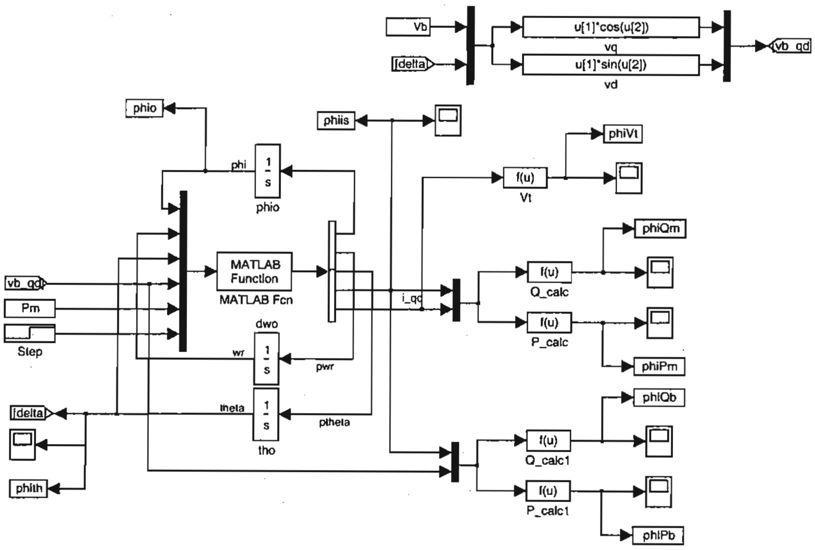


Figure 6.12: Flux model generator.

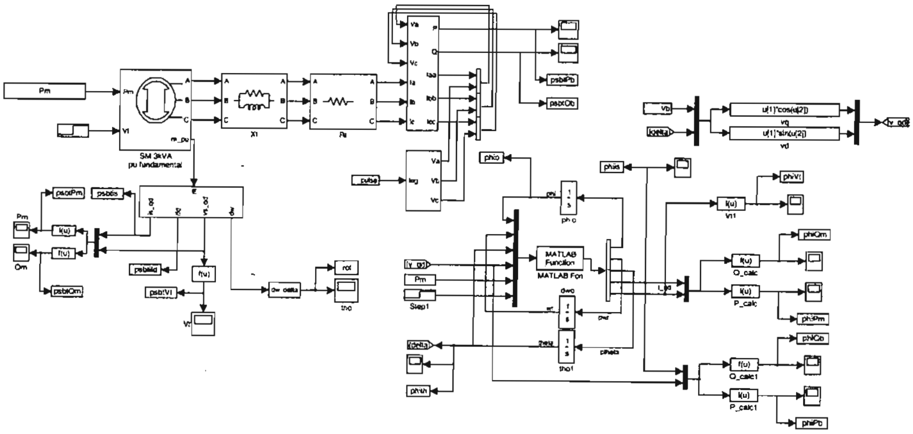


Figure 6.13: Simulation model for comparing the flux-model against the PSB model.

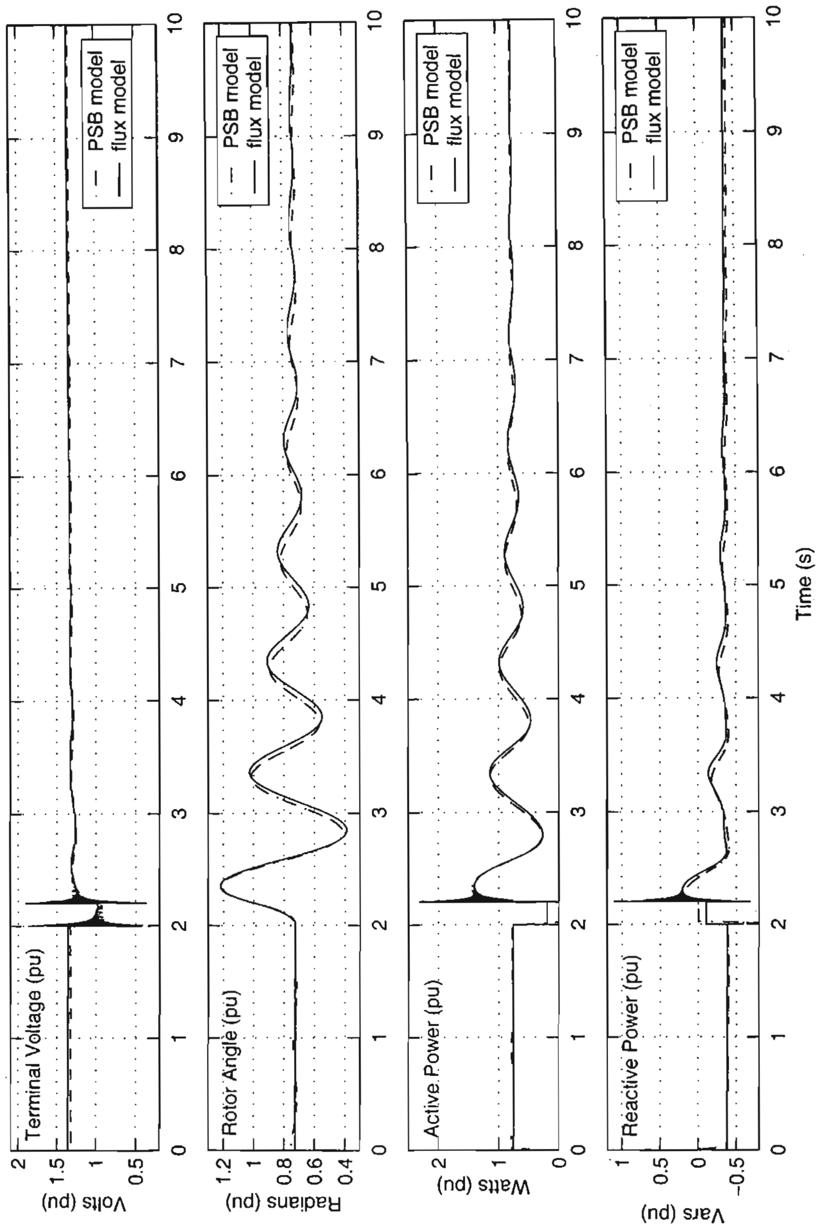


Figure 6.14: The result of the comparison between the flux and PSB models.

Chapter 7

Application of QFT to PSS Design

> He who loves practice without theory is like the sailor who boards ship without a rudder and compass and never knows where he may cast.
>
> LEONARDO DA VINCI

Chapter 6 described the physical model on which the QFT controller will be tested. Chapter 2 and Chapter 3 developed the mathematical and simulation model of an arbitrary power system. This chapter bridges the gap between the physical model and simulation model by integrating the physical model parameters into the simulation model, thus building a simulation environment that accurately represents the laboratory system. It also ties in Chapter 4 by developing a QFT controller for the simulation system. If the controller performs as expected in the simulation environment, there will be confidence that it will perform just as well in the physical model i.e the laboratory power system. The procedure is simple. First the mathematical model will be converted to a Power System Blockset model with all the physical model parameters. Secondly, the Simulink model will be converted to a suitable simulation model on which QFT design can be done. Thirdly, the QFT Toolbox will be used to design a controller. Finally, the designed controller will be simulated with the Power System Blockset model.

7.1 Linearised Models

It is possible to obtain frequency response data from the nonlinear model by simulating it for different input signal frequencies. This approach was discarded because of time constraints and flexibility. Linear models offer greater flexibility. They can be easily manipulated using the Control Toolbox [30] and simulation times are faster. The PSS is for small signal damping with unknown operating conditions. Sets of linearised models are appropriate and adequate for this design. The simulation models were linearised about a certain operating point. Since an infinite number of operating conditions exist, a general linearised model was developed for each power system component. A specific linearised model can be obtained by re-calculation of the linear equation coefficients.

7.1.1 Synchronous generator

The equations describing a synchronous machine in Chapter 2 are reproduced below for convenience.

$$v_d = -\, R_s i_d + \omega\lambda_q + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_d \tag{7.1}$$

$$v_q = -\, R_s i_q + \omega\lambda_d + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_q \tag{7.2}$$

$$v_{fd} = R_{fd} i_{fd} + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_{fd} \tag{7.3}$$

$$v_{kd} = R_{kd} i_{kd} + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_{kd} \tag{7.4}$$

$$v_{kq} = R_{kq} i_{kq} + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_{kq} \tag{7.5}$$

These are nonlinear equations, as used by the PSB. To solve these equations, it is useful to have the derivative variable on the right hand side, as shown in Equations 7.6–7.10.

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_d = v_d + R_s i_d - \omega\lambda_q \tag{7.6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_q = v_q + R_s i_q - \omega\lambda_d \tag{7.7}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_{fd} = v_{fd} - R_{fd} i_{fd} \tag{7.8}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_{kd} = v_{kd} - R_{kd} i_{kd} \tag{7.9}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_{kq} = v_{kq} - R_{kq} i_{kq} \tag{7.10}$$

The state variables are therefore λ_d, λ_q, λ_{fd}, λ_{kd}, λ_{kq}, ω, and θ. It is necessary to have the currents in terms of the state variables. This conversion is done via the following algebraic equations [21].

$$\lambda_d = -(L_{md} + L_l)i_d + L_{md}i_{fd} + L_{md}i_{kd} \tag{7.11}$$
$$\lambda_{fd} = (L_{md} + L_{lfd})i_{fd} - L_{md}i_d + L_{md}i_{kd} \tag{7.12}$$
$$\lambda_{kd} = (L_{md} + L_{lkd})i_{kd} - L_{md}i_d + L_{md}i_{fd} \tag{7.13}$$
$$\lambda_q = -(L_{mq} + L_l)i_q + L_{mq}i_{kq} \tag{7.14}$$
$$\lambda_{kq} = (L_{mq} + L_{lkq})i_{kq} - L_{mq}i_q \tag{7.15}$$

The *d-axis* current in terms of flux linkages are

$$\begin{aligned}\begin{bmatrix}\lambda_d\\ \lambda_{fd}\\ \lambda_{kd}\end{bmatrix} &= \begin{bmatrix}-(L_{md}+L_l) & L_{md} & L_{md}\\ -L_{md} & (L_{md}+L_{lfd}) & L_{md}\\ -L_{md} & L_{md} & (L_{md}+L_{lkd})\end{bmatrix}\begin{bmatrix}i_d\\ i_{fd}\\ i_{kd}\end{bmatrix}\\ &= \boldsymbol{L_d}\begin{bmatrix}i_d\\ i_{fd}\\ i_{kd}\end{bmatrix} \qquad (7.16)\\ \Rightarrow \begin{bmatrix}i_d\\ i_{fd}\\ i_{kd}\end{bmatrix} &= \boldsymbol{L_d}^{-1}\begin{bmatrix}\lambda_d\\ \lambda_{fd}\\ \lambda_{kd}\end{bmatrix}\end{aligned}$$

The *q-axis* current in terms of flux linkages are

$$\begin{aligned}\begin{bmatrix}\lambda_q\\ \lambda_{kq}\end{bmatrix} &= \begin{bmatrix}-(L_{mq}+L_l) & L_{mq}\\ -L_{mq} & (L_{mq}+L_{lkq})\end{bmatrix}\begin{bmatrix}i_q\\ i_{kq}\end{bmatrix}\\ &= \boldsymbol{L_q}\begin{bmatrix}i_q\\ i_{kq}\end{bmatrix} \qquad (7.17)\\ \Rightarrow \begin{bmatrix}i_q\\ i_{kq}\end{bmatrix} &= \boldsymbol{L_q}^{-1}\begin{bmatrix}\lambda_q\\ \lambda_{kq}\end{bmatrix}\end{aligned}$$

Substituting these currents into Equations 7.6-7.10 gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_d = v_d + (1+\omega)\lambda_q + \frac{R_s}{L_l}(\lambda_{md} - \lambda_d) \tag{7.18}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_q = v_q - (1+\omega)\lambda_d + \frac{R_s}{L_l}(\lambda_{mq} - \lambda_q) \tag{7.19}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_{fd} = v_{fd}\frac{R_{fd}}{L_{md}} + \frac{R_{fd}}{L_{lfd}}(\lambda_{md} - \lambda_{fd}) \tag{7.20}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_{kd} = \frac{R_{kd}}{L_{lkd}}(\lambda_{md} - \lambda_{kd}) \tag{7.21}$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\lambda_{kq} = \frac{R_{kq}}{L_{lkq}}(\lambda_{mq} - \lambda_{kq}) \tag{7.22}$$

The damper winding voltages v_{kd} and v_{kq} are assumed to be zero, and the restriction on the equations is that the parameter values are in per unit.

λ_{mq} and λ_{md}, the mutual flux linkages, are defined by

$$\lambda_{md} = L_{ad}\left(\frac{\lambda_d}{L_l} + \frac{\lambda_{fd}}{L_{lfd}} + \frac{\lambda_{kd}}{L_{lkd}}\right) \tag{7.23}$$

$$\lambda_{mq} = L_{aq}\left(\frac{\lambda_q}{L_l} + \frac{\lambda_{kq}}{L_{lkq}}\right) \tag{7.24}$$

with L_{ad} and L_{aq} given by

$$L_{ad} = \left(\frac{1}{L_l} + \frac{1}{L_{lfd}} + \frac{1}{L_{lkd}}\right)^{-1} \tag{7.25}$$

$$L_{aq} = \left(\frac{1}{L_l} + \frac{1}{L_{lkq}}\right)^{-1} \tag{7.26}$$

Now that there is confidence that the nonlinear flux-linkage model gives the same output as the PSB model as confirmed in Chapter 6.11, the flux-linkage equations can be linearised.

The flux-linkage equations given by Equations 7.18–7.22 may be linearised by either using the Jacobian matrix, or by applying a small disturbance Δx to the steady state value x_0, where x is the variable under consideration. The Jacobian is an elegant and compact method, and it is used here. The flux-linkage equations are linearised as follows. Equations 7.18–7.22 can be rewritten as

$$\begin{aligned}
\dot{\lambda}_1 &= f_1\left(\lambda_d, \lambda_q, \lambda_{fd}, \lambda_{kd}, \lambda_{kq}, v_d\right)\\
\dot{\lambda}_2 &= f_2\left(\lambda_d, \lambda_q, \lambda_{fd}, \lambda_{kd}, \lambda_{kq}, v_q\right)\\
\dot{\lambda}_3 &= f_3\left(\lambda_d, \lambda_q, \lambda_{fd}, \lambda_{kd}, \lambda_{kq}, v_{fd}\right)\\
\dot{\lambda}_4 &= f_4\left(\lambda_d, \lambda_q, \lambda_{fd}, \lambda_{kd}, \lambda_{kq}, 0\right)\\
\dot{\lambda}_5 &= f_5\left(\lambda_d, \lambda_q, \lambda_{fd}, \lambda_{kd}, \lambda_{kq}, 0\right)
\end{aligned} \tag{7.27}$$

In vector notation, this is

$$\dot{\boldsymbol{\lambda}} = \boldsymbol{f}(\boldsymbol{\lambda}, \boldsymbol{v}) \tag{7.28}$$

where

$$\begin{aligned}
\boldsymbol{\lambda} &= \begin{bmatrix}\lambda_d & \lambda_q & \lambda_{fd} & \lambda_{kd} & \lambda_{kq}\end{bmatrix}^T\\
\boldsymbol{v} &= \begin{bmatrix}v_d & v_q & v_{fd} & 0 & 0\end{bmatrix}^T
\end{aligned} \tag{7.29}$$

Applying the vector Taylor series expansion to $\dot{\boldsymbol{\lambda}}$ gives

$$\boldsymbol{f}(\boldsymbol{\lambda}, \boldsymbol{v}) \approx \boldsymbol{f}(\boldsymbol{\lambda}_0, \boldsymbol{v}_0) + \frac{\partial \boldsymbol{f}(\boldsymbol{\lambda}_0, \boldsymbol{v}_0)}{\partial \boldsymbol{\lambda}}\left(\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\right) + \frac{\partial \boldsymbol{f}(\boldsymbol{\lambda}_0, \boldsymbol{v}_0)}{\partial \boldsymbol{v}}\left(\boldsymbol{v} - \boldsymbol{v}_0\right) \tag{7.30}$$

where the Jacobian matrices are

$$\frac{\partial \boldsymbol{f}(\boldsymbol{\lambda_0}, \boldsymbol{v_0})}{\partial \boldsymbol{\lambda}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial \lambda_d} & \frac{\partial f_1}{\partial \lambda_q} & \frac{\partial f_1}{\partial \lambda_{fd}} & \frac{\partial f_1}{\partial \lambda_{kd}} & \frac{\partial f_1}{\partial \lambda_{kq}} \\ \frac{\partial f_2}{\partial \lambda_d} & \frac{\partial f_2}{\partial \lambda_q} & \frac{\partial f_2}{\partial \lambda_{fd}} & \frac{\partial f_2}{\partial \lambda_{kd}} & \frac{\partial f_2}{\partial \lambda_{kq}} \\ \frac{\partial f_3}{\partial \lambda_d} & \frac{\partial f_3}{\partial \lambda_q} & \frac{\partial f_3}{\partial \lambda_{fd}} & \frac{\partial f_3}{\partial \lambda_{kd}} & \frac{\partial f_3}{\partial \lambda_{kq}} \\ \frac{\partial f_4}{\partial \lambda_d} & \frac{\partial f_4}{\partial \lambda_q} & \frac{\partial f_4}{\partial \lambda_{fd}} & \frac{\partial f_4}{\partial \lambda_{kd}} & \frac{\partial f_4}{\partial \lambda_{kq}} \\ \frac{\partial f_5}{\partial \lambda_d} & \frac{\partial f_5}{\partial \lambda_q} & \frac{\partial f_5}{\partial \lambda_{fd}} & \frac{\partial f_5}{\partial \lambda_{kd}} & \frac{\partial f_5}{\partial \lambda_{kq}} \end{bmatrix}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda_0},\, \boldsymbol{v}=\boldsymbol{v_0}} \tag{7.31}$$

and

$$\frac{\partial \boldsymbol{f}(\boldsymbol{\lambda_0}, \boldsymbol{v_0})}{\partial \boldsymbol{\lambda}} = \left.\begin{bmatrix} \frac{\partial f_1}{\partial v_d} & \frac{\partial f_1}{\partial v_q} & \frac{\partial f_1}{\partial v_{fd}} & 0 & 0 \\ \frac{\partial f_2}{\partial v_d} & \frac{\partial f_2}{\partial v_q} & \frac{\partial f_2}{\partial v_{fd}} & 0 & 0 \\ \frac{\partial f_3}{\partial v_d} & \frac{\partial f_3}{\partial v_q} & \frac{\partial f_3}{\partial v_{fd}} & 0 & 0 \\ \frac{\partial f_4}{\partial v_d} & \frac{\partial f_4}{\partial v_q} & \frac{\partial f_4}{\partial v_{fd}} & 0 & 0 \\ \frac{\partial f_5}{\partial v_d} & \frac{\partial f_5}{\partial v_q} & \frac{\partial f_5}{\partial v_{fd}} & 0 & 0 \end{bmatrix}\right|_{\boldsymbol{\lambda}=\boldsymbol{\lambda_0},\, \boldsymbol{v}=\boldsymbol{v_0}} \tag{7.32}$$

For the nonlinear output equation

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{\lambda}, \boldsymbol{v}) \tag{7.33}$$

the linearisation is

$$\boldsymbol{y} - \boldsymbol{y_0} \approx \frac{\partial \boldsymbol{g}(\boldsymbol{\lambda_0}, \boldsymbol{v_0})}{\partial \boldsymbol{\lambda}}(\boldsymbol{\lambda} - \boldsymbol{\lambda_0}) + \frac{\partial \boldsymbol{g}(\boldsymbol{\lambda_0}, \boldsymbol{v_0})}{\partial \boldsymbol{v}}(\boldsymbol{v} - \boldsymbol{v_0}) \tag{7.34}$$

The linearised equations can be used to generate a state-space model with the *ABCD* matrices available for global processing in Simulink. The program given in Appendix F.15 is used to generate the state space model.

7.1.2 Transmission line

The transmission line is modelled in d–q coordinates and is described by the following equations

$$\lambda_d = (L_t - M)i_d \tag{7.35}$$

$$\lambda_q = (L_t - M)i_q \tag{7.36}$$

$$v_{td} = R_t i_d + \omega\lambda_q + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_d \tag{7.37}$$

$$v_{tq} = R_t i_q + \omega\lambda_d + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_q \tag{7.38}$$

The d–q representation of an infinite bus is $v_{bd} = v_b \cos\theta$ and $v_{bq} = v_b \sin\theta$, where the magnitude is given by $v_b = \sqrt{v_{bd}^2 + v_{bq}^2}$. For a transmission line connected to an infinite bus, the voltage equation terms become $v_d = v_{td} + v_{bd}$ and $v_q = v_{tq} + v_{bq}$. M is the inductive coupling between the phases of the transmission line and it is usually taken as zero. Therefore $i_d = \frac{\lambda_d}{L_t}$ and $i_q = \frac{\lambda_q}{L_t}$. The voltage equations then become

$$v_d = \frac{R_t}{L_t}\lambda_d + \omega\lambda_q + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_d + v_b\cos\theta \tag{7.39}$$

$$v_q = \frac{R_t}{L_t}\lambda_q + \omega\lambda_d + \frac{\mathrm{d}}{\mathrm{d}t}\lambda_q + v_b\sin\theta \tag{7.40}$$

These equations have nonlinear behaviour in ω and λ. They can be linearised by applying the Taylor expansion, e.g. for v_q,

$$\begin{aligned}(\Delta v_q + v_{q0}) = \frac{R_t}{L_t}(\Delta\lambda_q + \lambda_{q0}) + (\Delta\omega + \omega_0)(\Delta\lambda + \lambda_{d0})\\ + \frac{\mathrm{d}}{\mathrm{d}t}(\Delta\lambda_q + \lambda_{q0}) + v_b\sin(\Delta\theta + \theta_0)\end{aligned} \tag{7.41}$$

Ignoring the steady state constants and second or higher terms, the small signal equation becomes

$$\Delta v_q = \frac{R_t}{L_t}\Delta\lambda_q + \lambda_{d0}\Delta\omega + \omega_0\Delta\lambda_d + \frac{\mathrm{d}}{\mathrm{d}t}\Delta\lambda_q + v_b\cos(\theta_0)\Delta\theta \tag{7.42}$$

Similarly,

$$\Delta v_d = \frac{R_t}{L_t}\Delta\lambda_d + \lambda_{q0}\Delta\omega + \omega_0\Delta\lambda_q + \frac{\mathrm{d}}{\mathrm{d}t}\Delta\lambda_d + v_b\sin(\theta_0)\Delta\theta \tag{7.43}$$

These equations are substituted in the linearised flux-linkage equations to give a linearised model of a single machine infinite bus system.

To test the validity of the linearised equations, it is compared to the nonlinear flux-linkage model given in Appendix F.6, at a particular operating point. A small disturbance is applied to both models, and the outputs of both models are sent to the workspace for comparison. The Simulink model is shown in Figure 7.1, and the results are shown in Figure 7.2.

It can be seen that the linear model does in fact give the same output as the flux-linkage model for small disturbances. Since the flux-linkage model had been verified with the PSB model, it can be said with certainty that the linear model is a linearisation of the nonlinear PSB model, at a particular operating point.

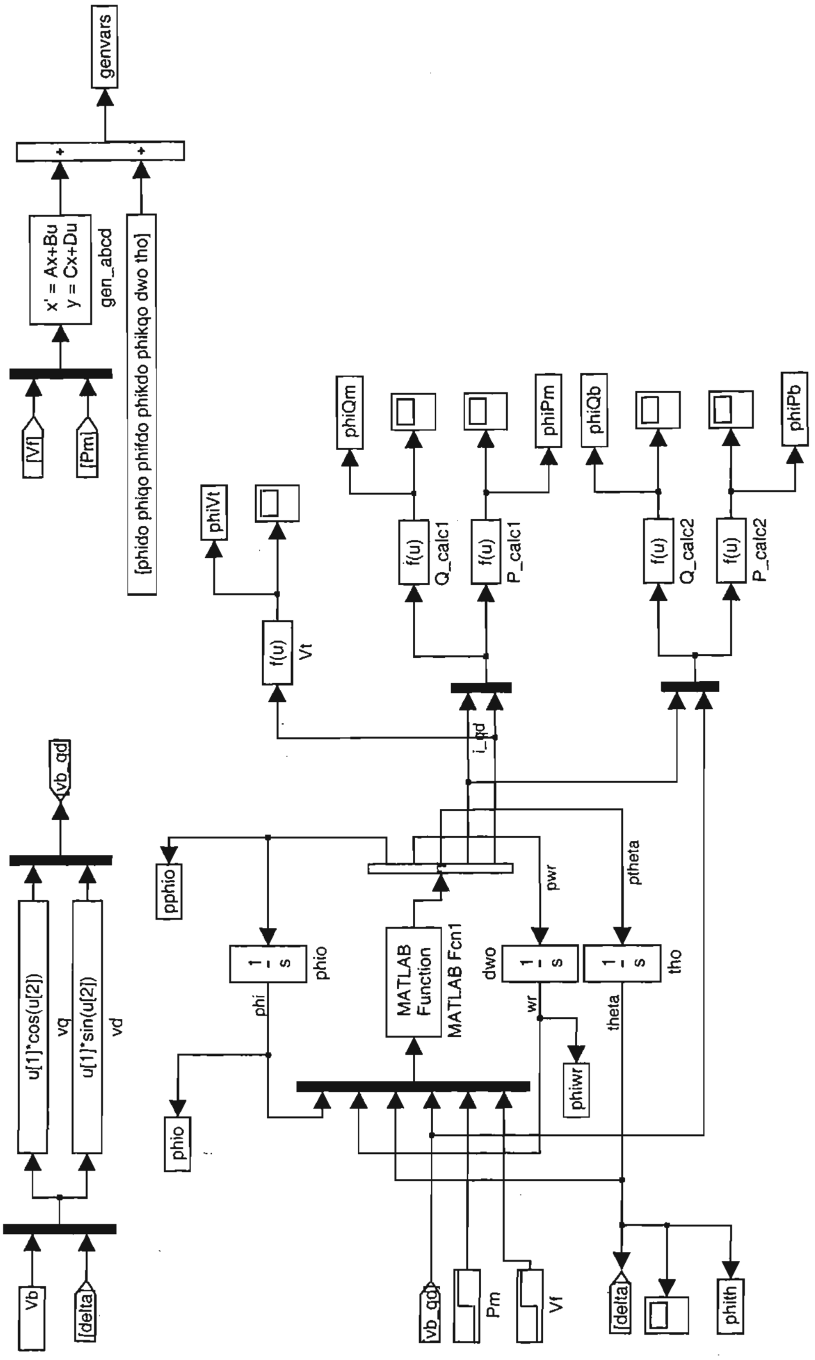


Figure 7.1: Simulink model to verify the linearised model against the flux model.

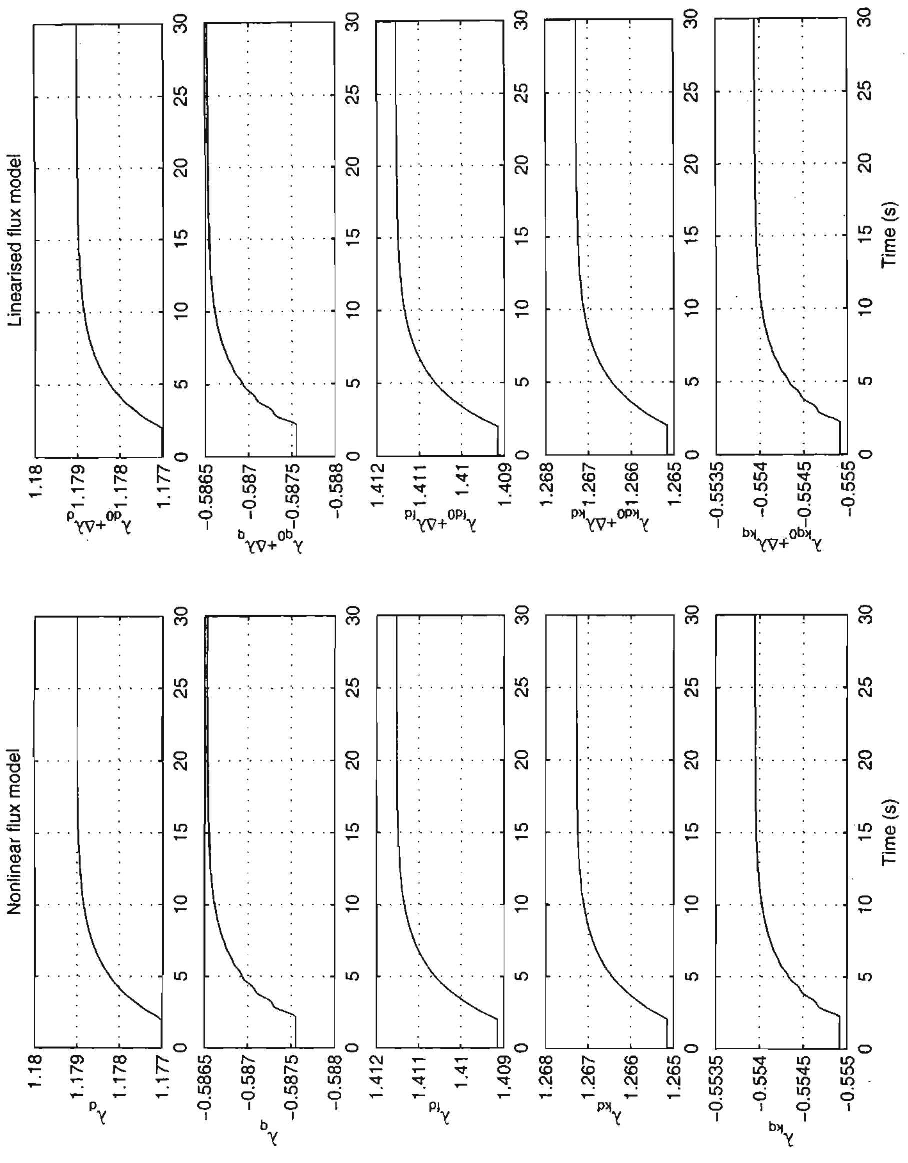


Figure 7.2: The result of the verification of the two models.

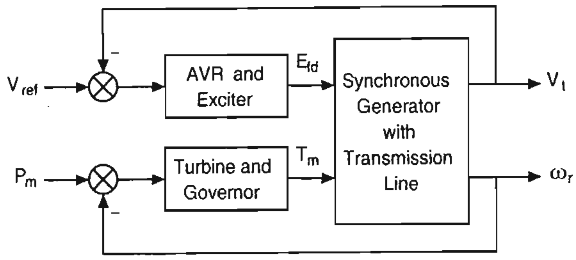


Figure 7.3: The power system setup.

7.2 Case Study 1: A first-order AVR

A state-space model is generated by assuming a first order AVR (not the laboratory AVR) given by the following equation

$$A(s) = \frac{200}{(0.02s + 1)} \tag{7.44}$$

This will allow the system to have a fast response. The governor model of the laboratory governor will be used, as well as the linearised generator model of the laboratory alternator. The power system block diagram is shown in Figure 7.3. Since the models of the components of the power system are linear time invariant (LTI) models, the Control System Toolbox in Matlab can be used to generate a power system LTI model. The AVR and governor are added in parallel and attached in series to the governor model. Matlab automatically generates the resulting inputs and outputs. The program that calculates and generates the entire power system model in state space is given in Appendix F.4. Detailed information on LTI model generation and manipulation can be found in [30]. Once the LTI system has been built, QFT design can commence.

7.2.1 QFT design

Using the LTI model of the generator connected to an infinite bus via a transmission line as a template, a set of LTI models are generated for different operating points and line reactance. These models are stored in a `m×n×p` matrix where p indexes a specific model. The LTI models are generated by calculating the steady state values of the state variables for a particular operating point. The m-file in Appendix F.2 does these calculations. In order to determine the frequency range of interest for which QFT design is to be done, a magnitude and phase plot of all the LTI systems is carried out. Figures 7.4 and 7.5 shows the magnitude and phase plot (bode diagram) of all the models.

Each LTI model represents a plant in the uncertainty region of active power

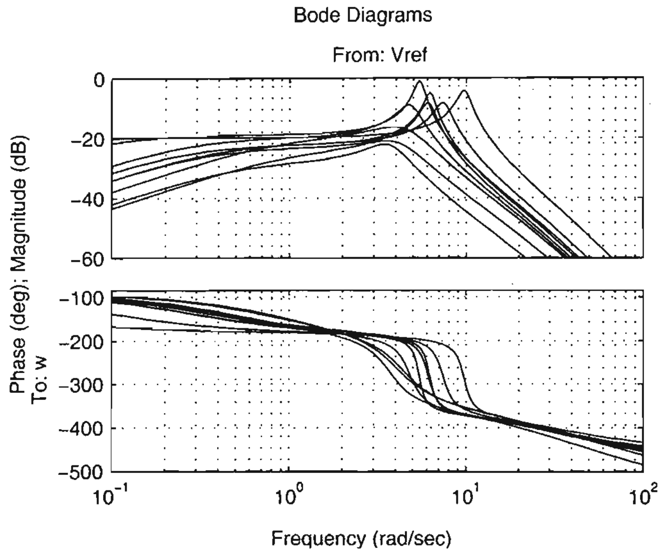


Figure 7.4: Bode plots of all models from V_{ref} to ω.

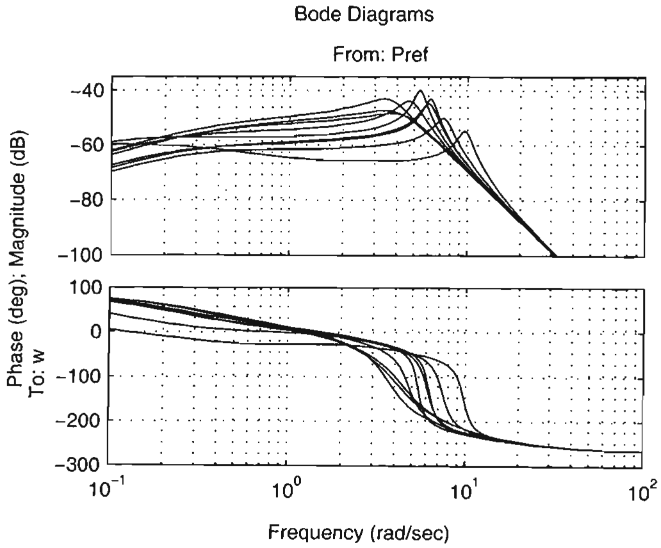


Figure 7.5: Bode plots of all models from P_{ref} to ω.

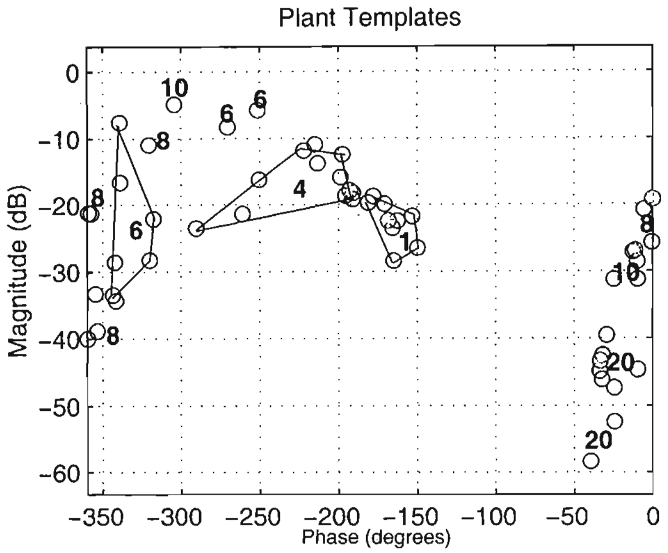


Figure 7.6: Plant templates for the different power system models.

P_b, reactive power Q_b, line reactance X_t, and line resistance R_t. It can be seen that the frequencies of interest are between $3\,\mathrm{rad\,s^{-1}}$ and $11\,\mathrm{rad\,s^{-1}}$, which is around the peak of the bode plot. A frequency vector is chosen which includes these points, and the plant templates for each frequency is generated. Figure 7.6 shows the plant templates. The numbers on the diagram indicate the frequencies at which the plant templates were calculated. The control specifications should be defined at this stage. For this design, the specifications are

1. that the power system be robust stable for all operating conditions and line reactance,

2. satisfy the robust stability criterion

$$\left|\frac{1}{1+L}\right| \leq 3dB \quad \forall\, \omega \tag{7.45}$$

3. provide damping factor $\xi \approx 0.25$ in the speed loop i.e.

$$\left|\frac{t_{22}}{1+L}\right| \leq 6dB + |t_{22}(j0)|_{dB} \quad \forall\, \omega \tag{7.46}$$

where t_{22} is the transfer function from P_{ref} to ω. The term $|t_{22}(j0)|_{dB}$ is the dc component of the speed loop, and it is taken into account for the QFT design.

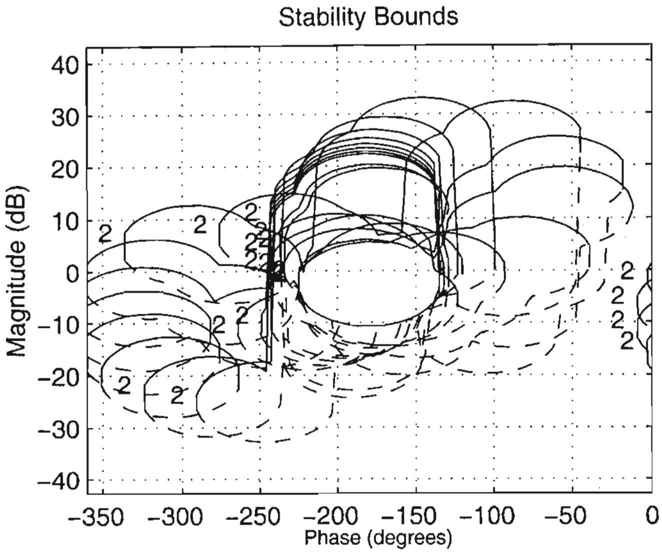


Figure 7.7: Stability bounds that are generated from the control requirements.

A plant having $P_b = 0.9$ p.u, $Q_b = 0.3$ p.u, $X_t = 0.542$ p.u, and $R_t = 0.0128$ p.u is chosen to be the nominal plant. The QFT Toolbox is used to calculate the stability bounds on the Nichols Chart for all the plant cases. The result is shown in Figure 7.7. The label "2" in Figure 7.7 indicates that the output disturbance rejection algorithm was used to calculate the bounds [9]. An initial controller of $G(s) = 1$ is chosen. Figure 7.9 shows the loop transmission curve with the initial controller.

The goal is to shape L_n such that it does not lie within the stability bounds. There are many ways to do this, therefore the solution is not unique. A possible solution is shown in Figure 7.10. The designed controller is given by Equation 7.47.

$$G(s) = 19.75\frac{(\frac{s}{24}+1)(\frac{s}{10}+1)}{(\frac{s}{141}+1)(\frac{s}{121}+1)} \tag{7.47}$$

One of the goals of loop shaping is to get a simple controller as possible and still obey the design specification.

7.2.2 Simulations and results

The LTI model with the PSS in place is verified by calculating the frequency response from the voltage reference input to the speed output. The result is shown in Figure 7.11. Figure 7.12 shows the frequency response from the mechanical power input to the speed output. The uncompensated system

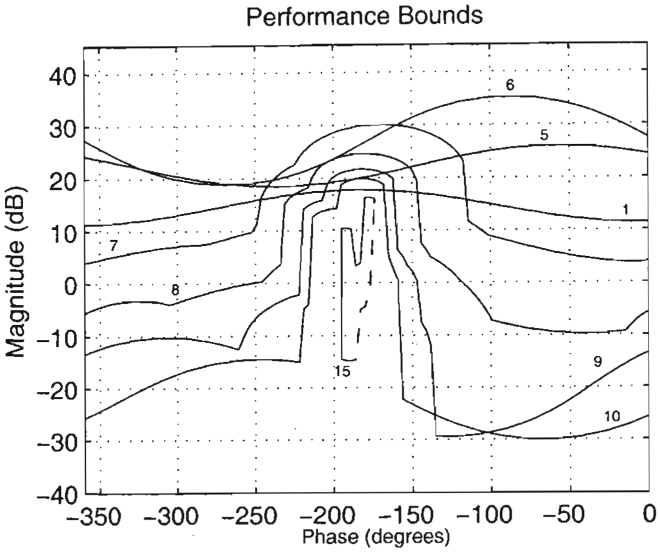


Figure 7.8: The performance bounds that are generated.

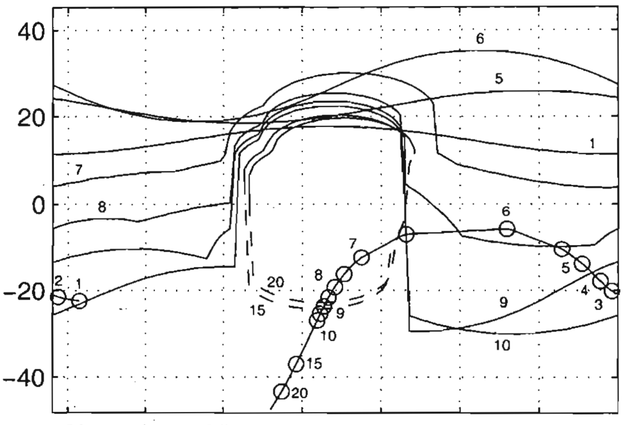


Figure 7.9: Start of QFT design with an initial controller $G(s)$.

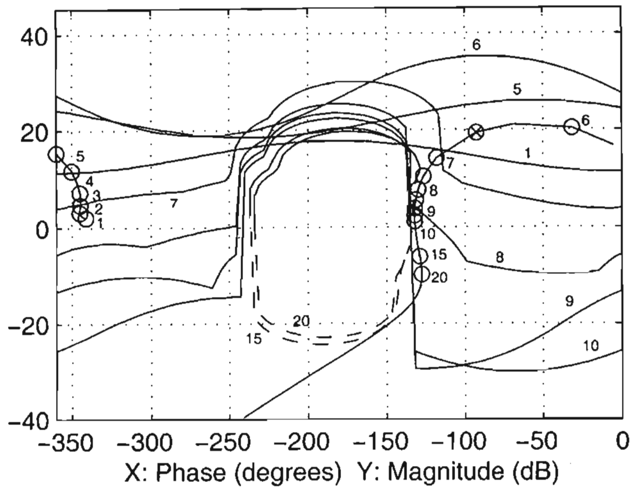


Figure 7.10: A QFT solution to meet the design specifications.

response is shown with a dotted line. The designed controller is also tested with the nonlinear model. The PSB is used for this task, and if need be, the large signal response can also be viewed. The single line electrical diagram is shown in Figure 7.13. Figure 7.14 illustrates the Simulink model with the controller connected in the feedback loop.

The small signal stability is also analysed. A step that is 5% of the voltage reference value is applied at time $t = 5$s. The response of the various power system quantities are shown in Figure 7.15. The plant for which the small signal stability response was calculated had $P_b = 0.9$ p.u, $Q_b = 0.3$ p.u, $X_t = 0.542$ p.u, and $R_t = 0.0142$ p.u. The dotted line shows the uncompensated response to the step input.

The large signal stability is also analysed. This time, there is no step input to the voltage reference input. Instead, a 3-phase short circuit to earth is simulated. This 3-phase fault is applied at point (A) in Figure 7.13 at time $t = 5$s. The uncompensated response is shown dashed in Figure 7.16, while the compensated response is the solid line.

Finally, the parametric stability is investigated. A lightly damped operating point is chosen ($P_b = 0.01$ p.u, $Q_b = 0.8$ p.u, $X_t = 0.542$ p.u, and $R_t = 0.0142$ p.u). The response of the power system at this operating condition is displayed in Figure 7.17, with dashed lines representing the uncompensated power system. The solid line shows the generator output with the power system stabiliser in place.

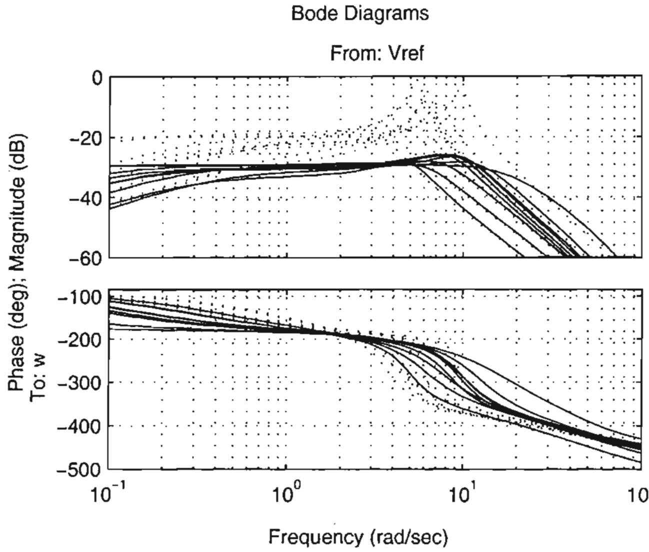


Figure 7.11: Bode plot of V_{ref} to ω with the PSS in place.

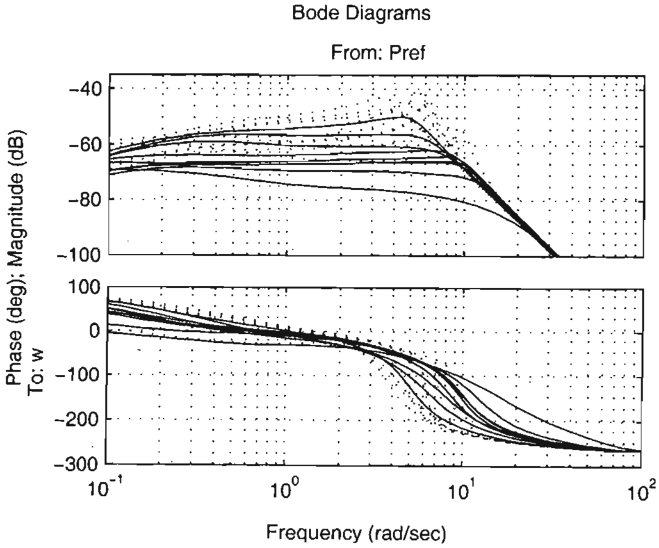


Figure 7.12: Bode plot of P_{ref} to ω with the PSS in place.

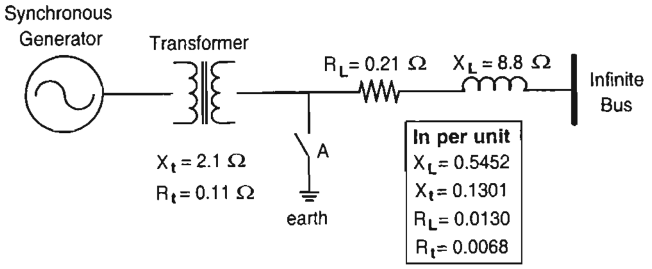


Figure 7.13: Diagram of the electrical network to be simulated.

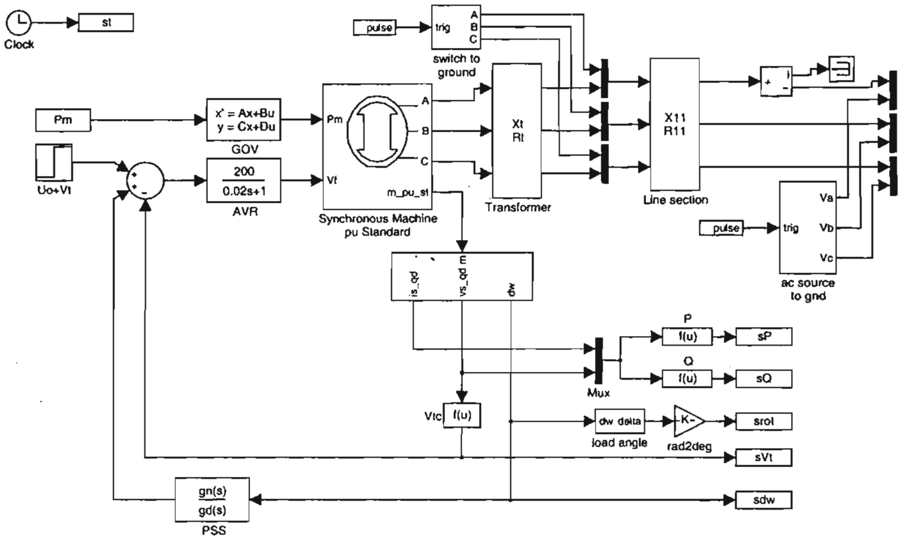


Figure 7.14: Nonlinear PSB model with the designed PSS.

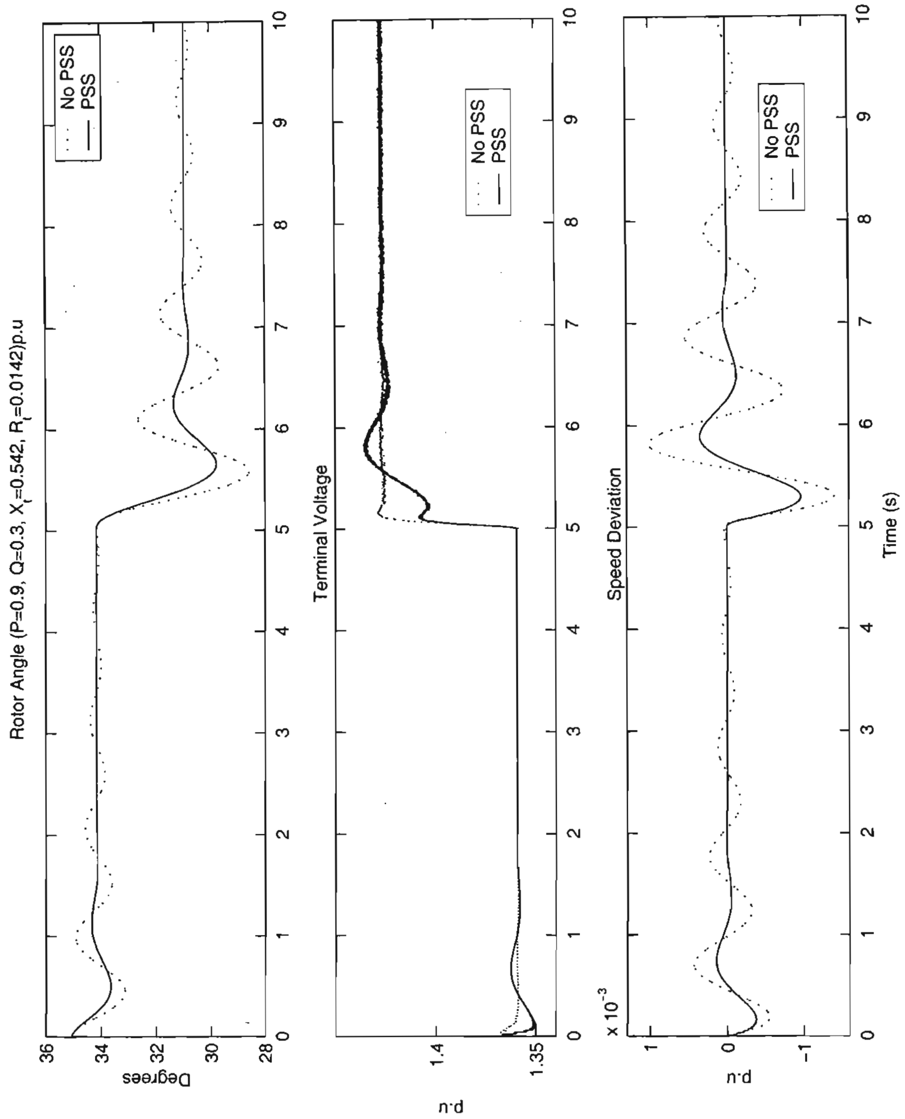


Figure 7.15: Result of the nonlinear small-signal disturbance test.

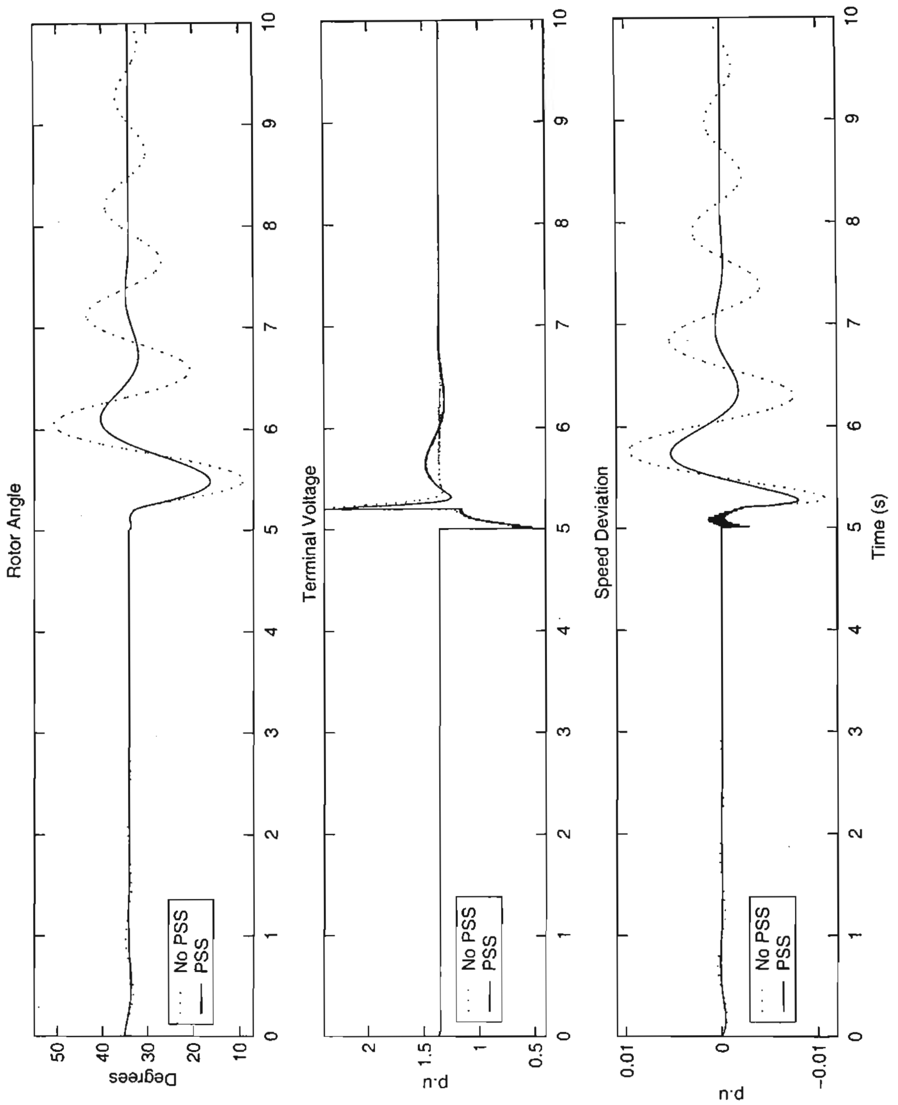


Figure 7.16: The nonlinear large-signal stability test.

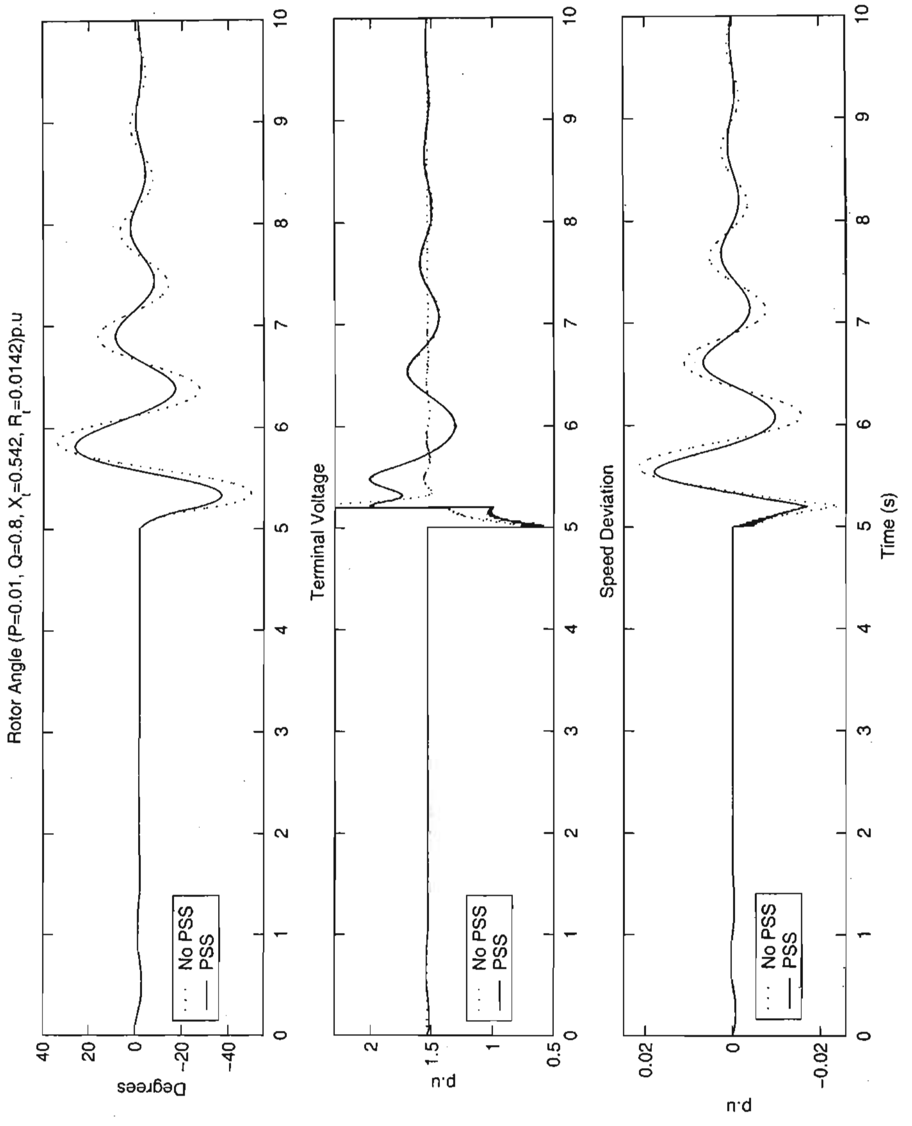


Figure 7.17: Result of a large-signal disturbance to the power system at a lightly damped operating point.

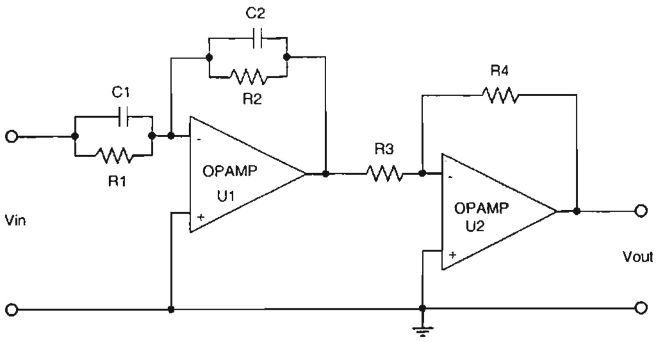


Figure 7.18: Electronic circuit representing the design equation.

7.2.3 Practical implementation

Once the controller has been designed and tested by computer simulation, the next step in the design process is to build the physical controller. For this, the controller must be proper, or preferably strictly proper. This consideration would have been in mind during the QFT design phase. The controller can be realised either in an analog, or digital, form. The analog realisation is chosen because of low component count, low cost, and its simplicity.

The circuit is built up using operational amplifiers, resistors, and capacitors. The design calculations for the resistor and capacitor values are as follows

1. the design equation
$$K_c \frac{s+\frac{1}{T}}{s+\frac{1}{\alpha T}} \tag{7.48}$$
where $T = R_1C_1$, $\alpha T = R_2C_2$, and $K_c = \frac{R_4C_1}{R_3C_2}$, is used for each zero-pole-gain pair in the controller transfer function. The electronic circuit representation of this design equation is shown in Figure 7.18,

2. there are 2 zero-pole-gain pairs in $G(s)$:
$$37\frac{s+24}{s+141} \quad \text{and} \tag{7.49}$$
$$37\frac{s+10}{s+121} \tag{7.50}$$

3. let $\frac{1}{T_1} = 24$, $\frac{1}{\alpha T_1} = 141$, $\frac{1}{T_2} = 10$, and $\frac{1}{\alpha T_2} = 121$,

4. choose $C_1 = 1\mu\text{F}$, $C_2 = 1\mu\text{F}$, $C_3 = 1\mu\text{F}$, and $C_4 = 1\mu\text{F}$,

5. the resistor values for the first transfer function, after the relevant calculations, are $R_1 = 40\text{k}\Omega$, $R_2 = 7\text{k}\Omega$, $R_3 = 1\text{k}\Omega$, and $R_4 = 37\Omega$, and

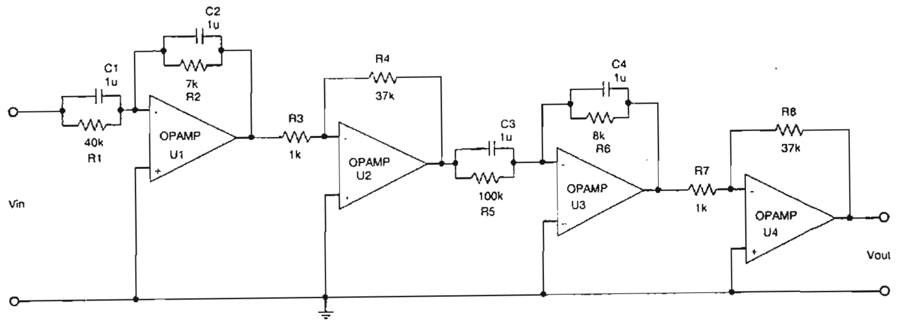


Figure 7.19: PSS circuit designed from $G(s)$.

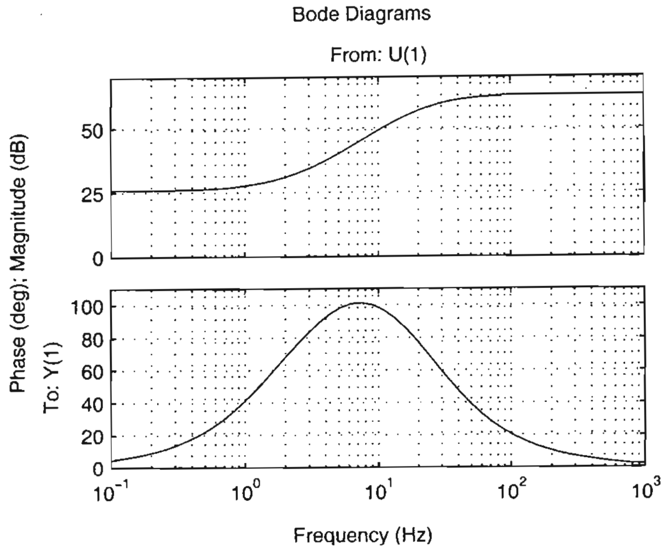


Figure 7.20: The bode plot of $G(s)$.

6. the second transfer function resistor values are R_5 = **100kΩ**, R_6 = **8kΩ**, R_7 = **1kΩ**, and R_4 = **37kΩ**.

The circuit was simulated in *PSPICE* before it is constructed. Figure 7.19 shows the power system stabiliser circuit.

The frequency response of the transfer function is shown in Figure 7.20. U(1) is the input to $G(s)$ and Y(1) is the output from $G(s)$. The frequency response of the PSS circuit is shown in Figure 7.21. This is the same as the frequency response of the controller transfer function. The designed circuit in Figure 7.19 is not used anywhere, and is merely used to highlight the simplicity of PSS design using QFT.

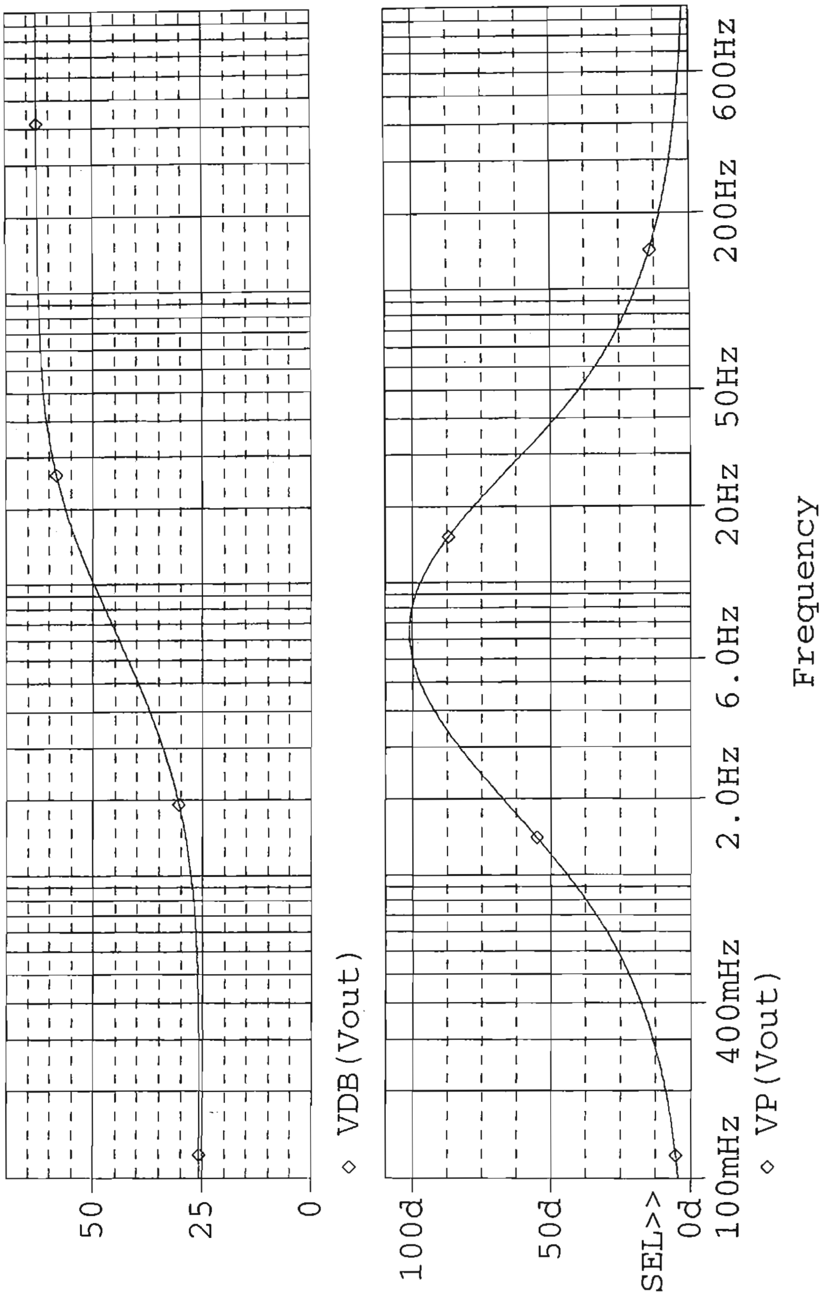


Figure 7.21: Bode plot of the designed electronic circuit.

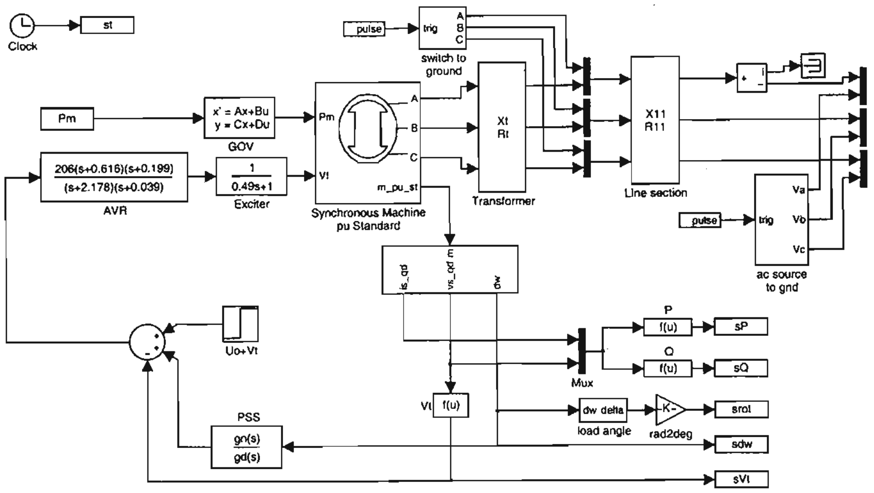


Figure 7.22: Simulink model with the laboratory AVR in place.

7.3 Case Study 2: Laboratory AVR

This case study involves the entire laboratory model i.e alternator model, AVR model, governor model, transmission line model, and infinite bus model. Once again, design is done on a decoupled SISO system. The reason for using different AVR models is that the AVR greatly influences the stability of a generator, as well as its response time. It is therefore preferable to design the power system stabiliser first, and then design the AVR such that the required phase and gain margins are achieved. However, most of the time, this is not possible, as power systems usually have the AVR as an integral part of their setup. For this case study, the first order AVR model is replaced by the laboratory AVR model. The Simulink model diagram is shown in Figure 7.22. The frequency responses from V_{ref} to ω and P_{ref} to ω for the same parametric set used in Case Study 1 is shown in Figures 7.23 and 7.24.

7.3.1 QFT design

The procedure is the same as for Case Study 1. The LTI models are calculated for different operating points and line reactance. The frequency vector is chosen such that the frequencies lie mostly in the region of interest as depicted in the bode plot. The plant templates are calculated for each frequency. The plant templates are plotted on the Nichols Chart. Figure 7.25 illustrates what the plant templates look like. The control specification for the controller is the same as before, i.e

1. that the power system be robust stable for all operating conditions

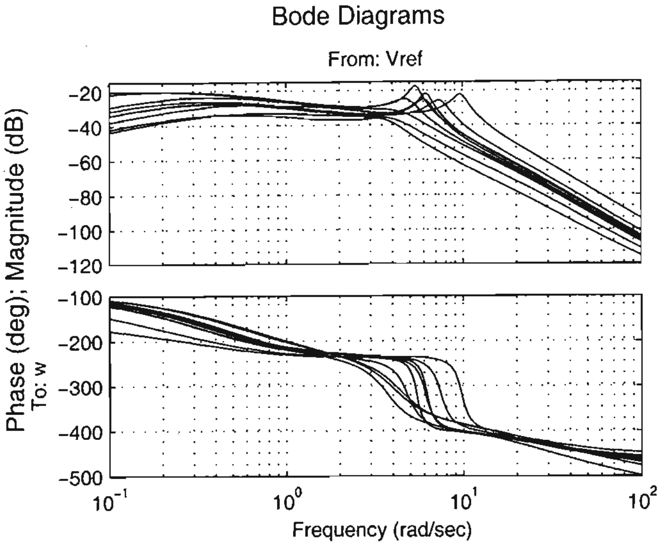


Figure 7.23: Bode plots from V_{ref} to ω for the power system plants.

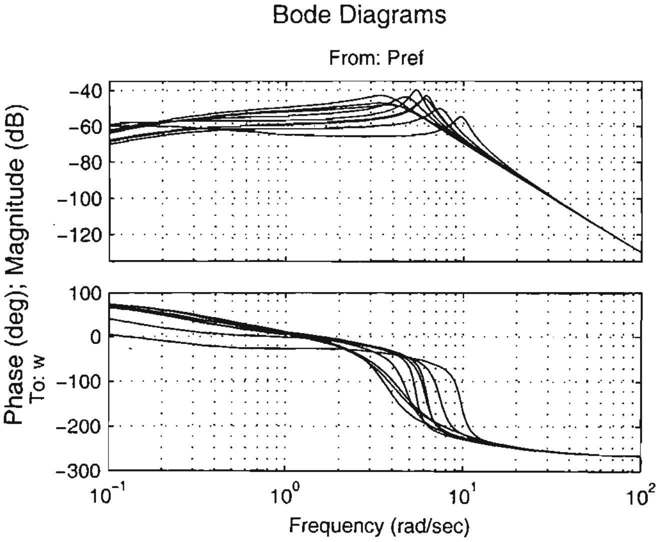


Figure 7.24: Bode plots from P_{ref} to ω for the power system plants.

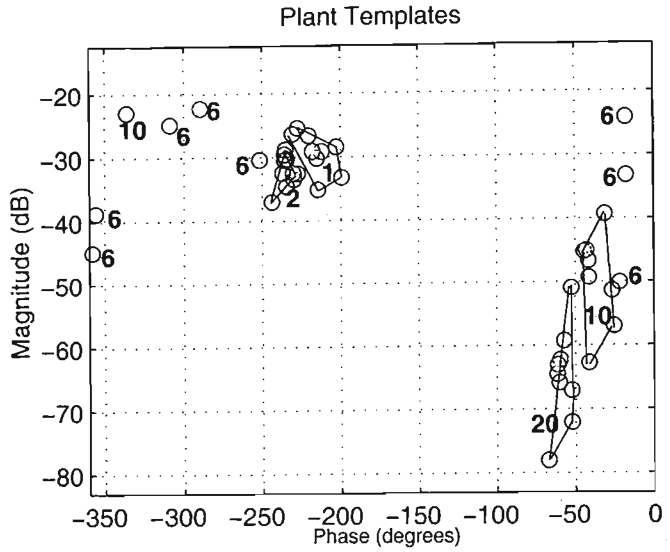


Figure 7.25: Plant templates for the parametric set of power system models.

and line reactances,

2. satisfy the robust stability criterion

$$\left|\frac{1}{1+L}\right| \leq 3dB \quad \forall\, \omega \tag{7.51}$$

$3dB$ is a reasonable estimate for the stability criterion, and gives some flexibility with loop shaping. A higher value would result in a more conservative design, and

3. provide damping factor $\xi \approx 0.25$ in the speed loop i.e.

$$\left|\frac{t_{22}}{1+L}\right| \leq 6dB + |t_{22}(j0)|_{dB} \quad \forall\, \omega \tag{7.52}$$

where t_{22} is the transfer function from P_{ref} to ω. $6dB$ is an initial design specification that may be changed to comply with the plant capabilities.

These specifications are discussed in detail in [7]. A plant having $P_b = 0.9$ p.u, $Q_b = 0.3$ p.u, $X_t = 0.542$ p.u, and $R_t = 0.0142$ p.u is chosen to be the nominal plant. Using these specifications, the stability bounds are calculated in Matlab. The programs given in Appendices F.1, F.3, F.4, and F.2 are used for the design. The loop shaping environment is invoked with a nominal plant referenced by index 1. An initial controller of $G(s) = \frac{s\tau_W}{(s\tau_W+1)}$

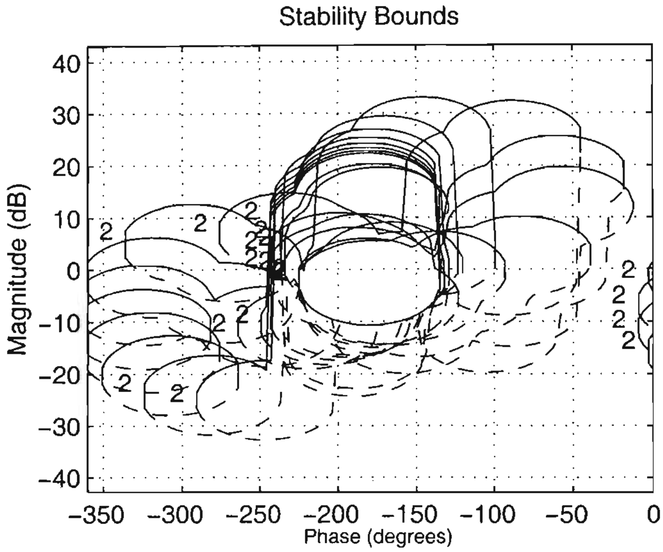


Figure 7.26: Stability bounds generated from the control specifications.

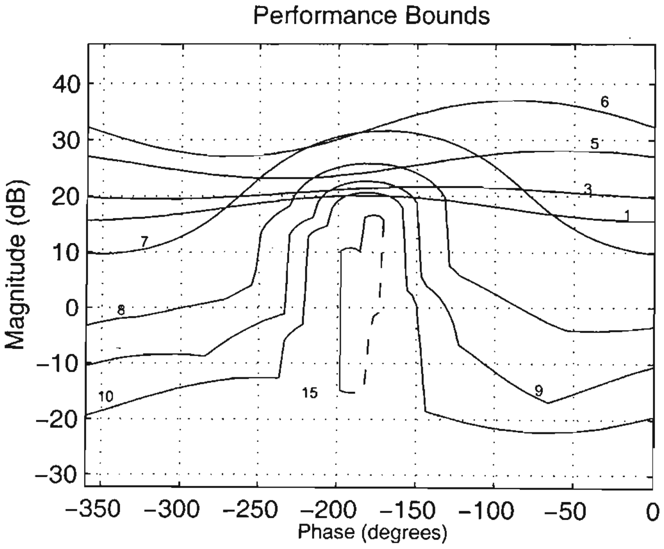


Figure 7.27: The performance bounds that the controller has to satisfy.

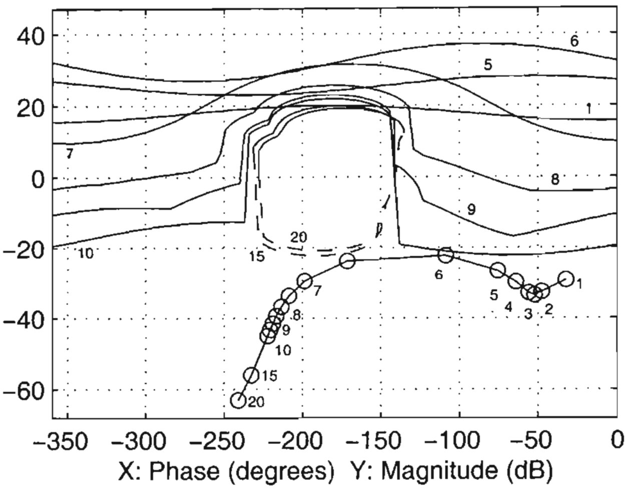


Figure 7.28: The interactive loop shaping environment with the initial controller $G(s)$.

is used. This initial controller is the washout term usually used in the design of power system stabilisers. The stability bounds, along with the loop transmission curve L_n, is displayed in Figure 7.28.

Loop shaping is carried out to meet the control specifications. Once again, there are an infinite number of solutions. A possible solution is shown in Figure 7.29.

The controller transfer function for the loop transmission is given by Equation 7.53.

$$G(s) = 2.48\frac{(s)(\frac{s}{1.44}+1)}{(\frac{s}{0.5}+1)(\frac{s}{28}+1)(\frac{s}{40}+1)} \tag{7.53}$$

7.3.2 Simulations and results

The controller is first tested by connecting it in the feedback loop of the nominal LTI power system model. Of course, other plants can also be used. Bode plots from V_{ref} to ω and P_{ref} to ω were generated with the PSS connected to the power system. The plant for which the frequency responses were calculated had $P_b = 0.9$ p.u, $Q_b = 0.3$ p.u, $X_t = 0.542$ p.u, and $R_t = 0.0128$ p.u. The results are shown in Figures 7.30 and 7.31.

The controller is then tested with the nonlinear model. The single line electrical diagram shown in Figure 7.13 is used for the simulation. A 5% step is applied to the AVR reference voltage input at a particular operating point. The step is applied at time $t = 5$. Figure 7.32 illustrates the uncompensated

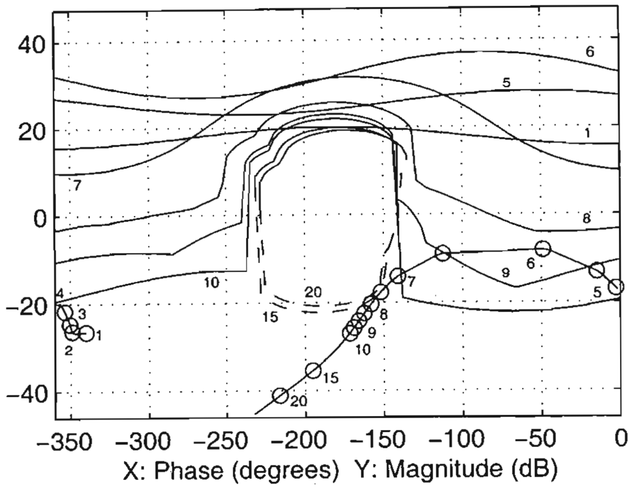


Figure 7.29: A QFT design solution to the control problem.

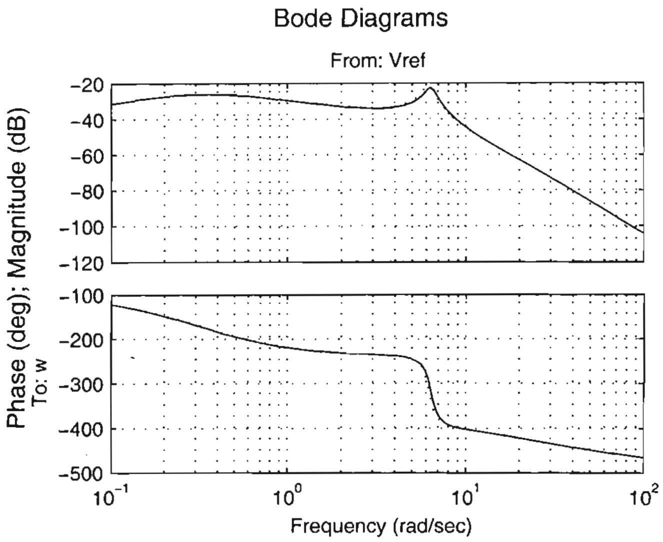


Figure 7.30: Bode plot of V_{ref} to ω with the PSS in place.

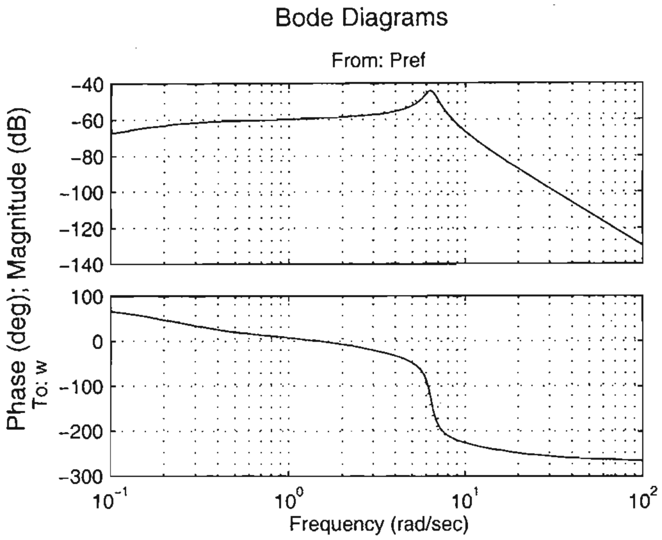


Figure 7.31: Bode plot of P_{ref} to ω with the PSS in place.

(dashed line) and compensated (solid line) responses of the power system model.

For large signal stability simulation, a 3-phase to earth fault is at applied at time $t = 5$. The fault is applied at point (A) in Figure 7.13. The outputs of the uncompensated and compensated system are shown in Figure 7.33. A parametric stability investigation at the same operating point as in Case Study 1 yields Figure 7.34.

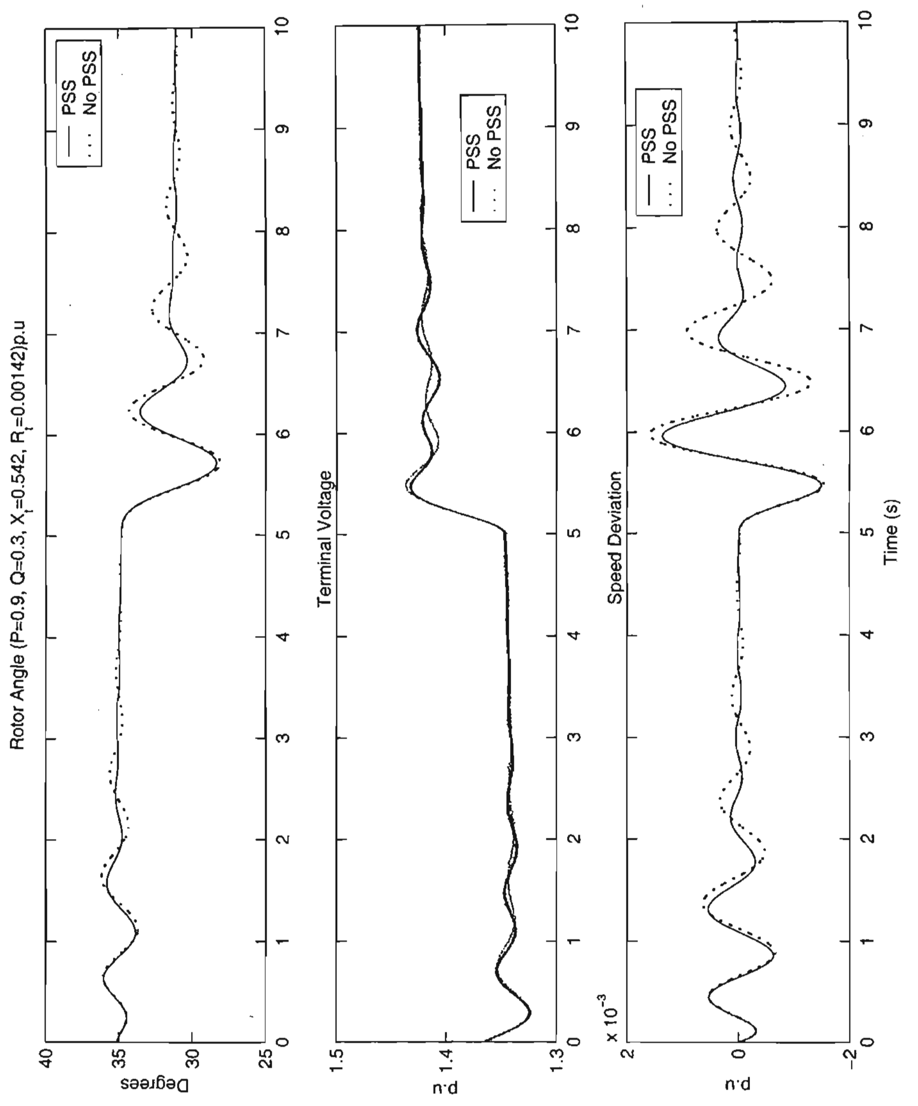


Figure 7.32: Small-signal disturbance response of the nonlinear model.

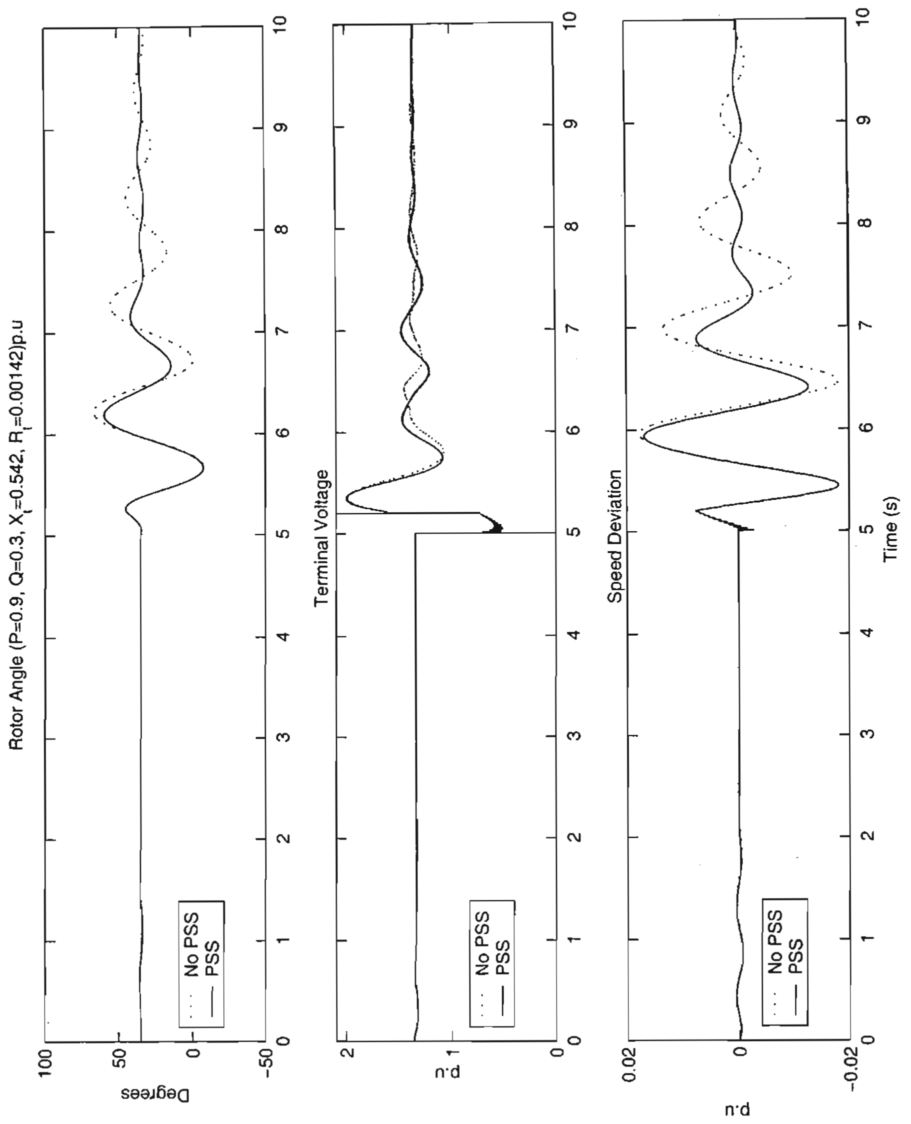


Figure 7.33: The result of a 3-phase fault to the nonlinear PSB model.

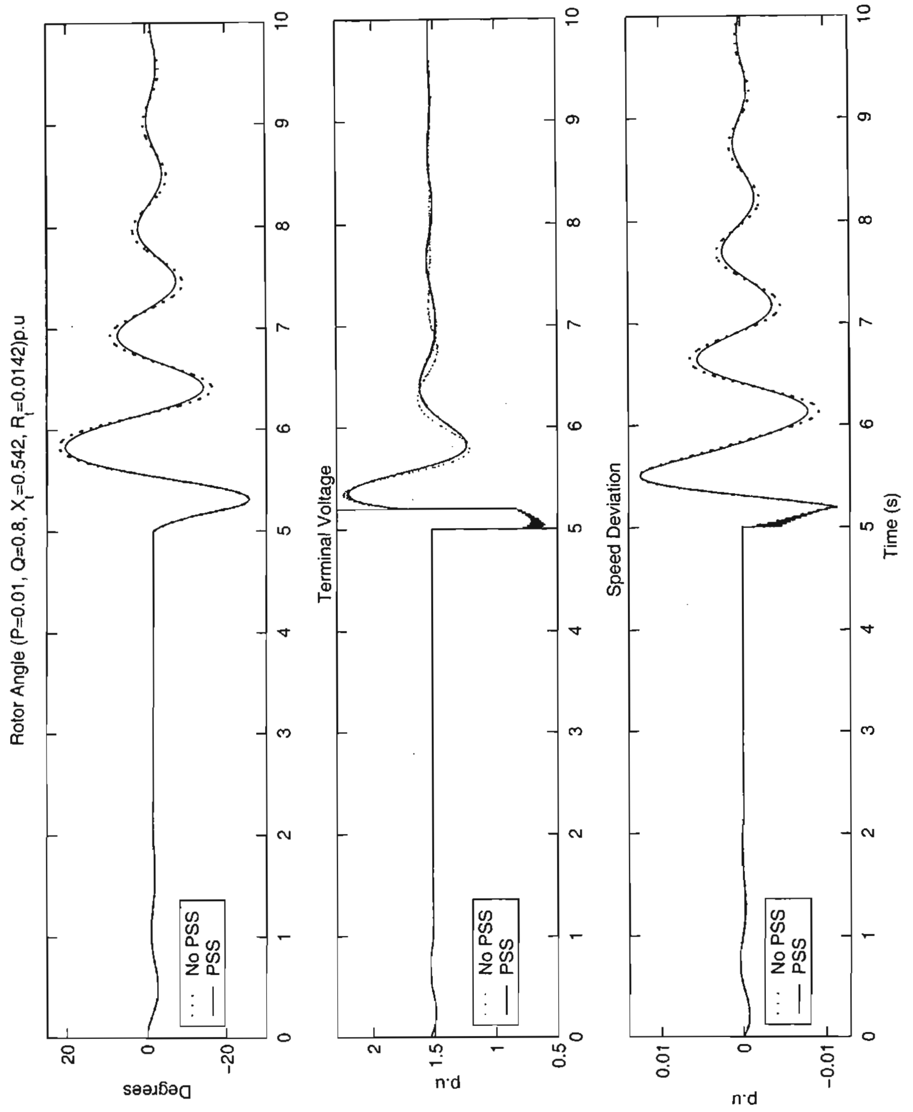


Figure 7.34: Result of a large-signal disturbance at a lightly damped operating point.

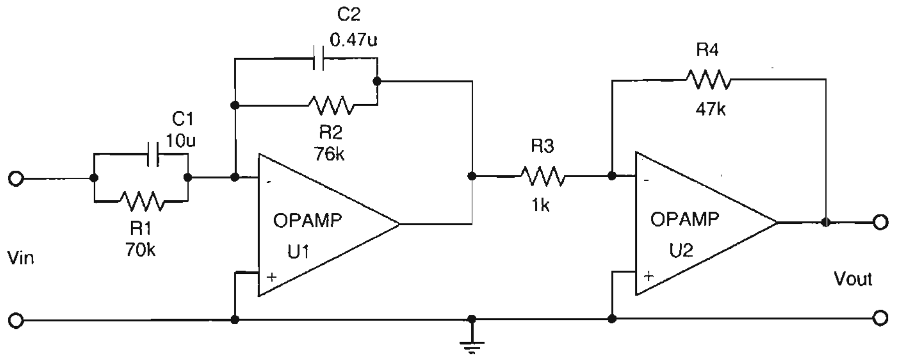


Figure 7.35: Electronic circuit for design equation $G_1(s)$.

7.3.3 Practical implementation

The proper controller can be realised without any problems. Once again, analog realisation is preferred. The design equations are based on the decomposition of the controller transfer function into single first order transfer functions which can be realised physically via operational amplifier circuits. See [32] for details.

1. rewrite $G(s)$ in terms of the individual first order transfer functions i.e.

$$G(s) = 962.9 \cdot \left(\frac{s+1.44}{s+28}\right) \cdot \left(\frac{1}{s+40}\right) \cdot \left(\frac{s}{s+0.5}\right) \tag{7.54}$$

2. taking the first transfer function, the design equation is

$$\begin{aligned} G_1(s) =& 962.9 \cdot \left(\frac{s+1.44}{s+28}\right) \\ =& \left(\frac{C_1 R_4}{C_2 R_3}\right) \cdot \left(\frac{s+\frac{1}{R_1 C_1}}{s+\frac{1}{R_2 C_2}}\right) \end{aligned} \tag{7.55}$$

   Choose $C_1 = 10\mu$F and $C_2 = 0.47\mu$F. Using the above equation, this gives $R_1 = 70$kΩ, $R_2 = 76$kΩ, $R_3 = 1$kΩ, and $R_4 = 47$kΩ. The equivalent electronic circuit for $G_1(s)$ is shown in Figure 7.35,

3. the design equation to realise the second transfer function is as follows

$$\begin{aligned} G_2(s) =& \frac{1}{40} \cdot \left(\frac{1}{0.025s+1}\right) \\ =& \left(-\frac{R_6}{R_5}\right) \cdot \left(\frac{1}{R_6 C_3 s+1}\right) \end{aligned} \tag{7.56}$$

   Choose $C_3 = 1\mu$F. This gives $R_5 = 1$MΩ and $R_6 = 25$kΩ. The design circuit to realise this transfer function is given in Figure 7.36, and

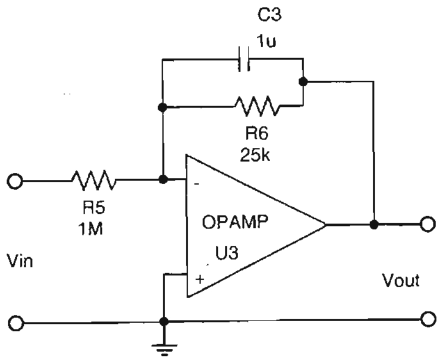


Figure 7.36: Electronic circuit for design equation $G_2(s)$.

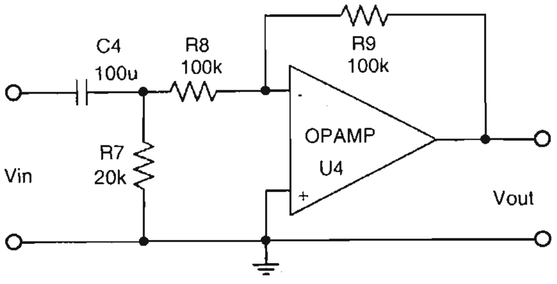


Figure 7.37: Electronic circuit for design equation $G_3(s)$.

4. the last transfer function can be realised by using a RC network followed by an inverting buffer. Therefore

$$\begin{aligned} G_3(s) &= \frac{s}{s+0.5} \\ &= -\frac{s}{s+\frac{1}{R_7C_4}} \end{aligned} \tag{7.57}$$

   From the design equation, $R_7C_4 = 2$. Choose $C_4 = 100\mu$F. This gives $R_7 = 20$kΩ. For the inverting buffer, choose $R_8 = R_9 = 100$kΩ. The circuit to realise this transfer function is shown in Figure 7.37.

Figure 7.38 shows the final power system stabiliser circuit design. This circuit will be built and used in the power system laboratory to test the QFT design process. Practical results from these tests can be found in the next chapter. Figure 7.39 shows the frequency response of the controller transfer function while Figure 7.40 shows the frequency response of the circuit.

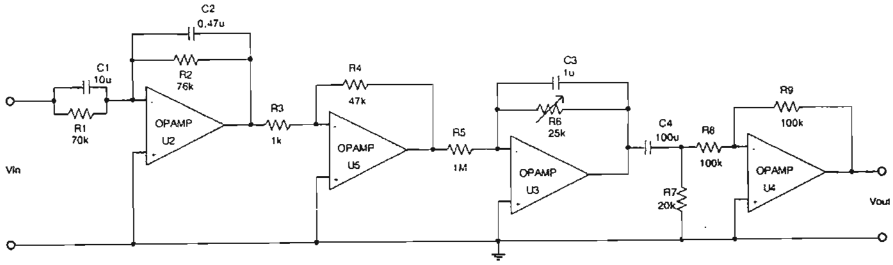


Figure 7.38: The designed analog PSS circuit.

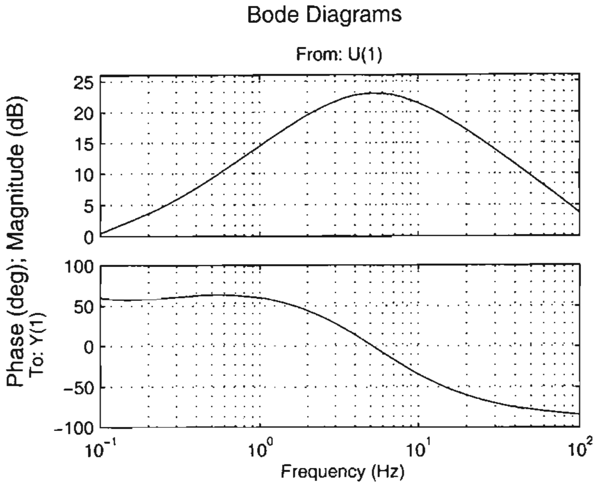


Figure 7.39: Frequency response of $G(s)$.

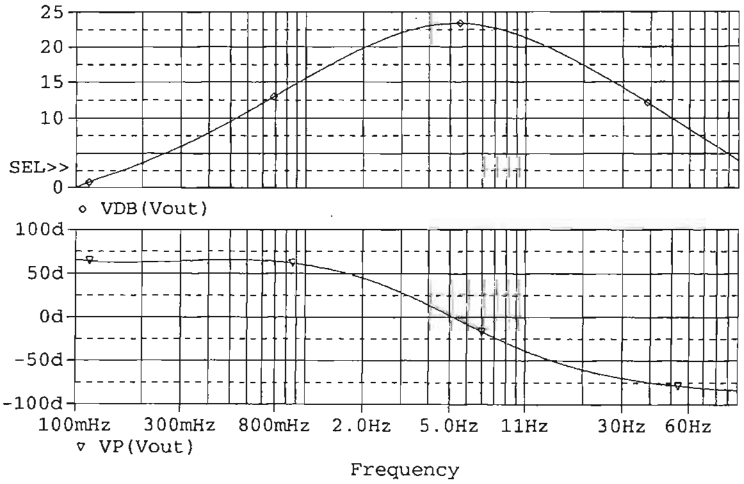


Figure 7.40: Frequency response of the electronic circuit.

7.4 Summary

QFT is the ideal tool to design power system stabilisers. It is easy to use. Solutions are transparent, and are quickly formed. QFT generates controllers that are robust stable and easy to implement, making it the perfect choice for industrial applications. The simulations verified that the designed controllers do indeed perform according to the control and performance specifications. The practical implementations showed how easily the designed controller can be synthesised.

The design required the use of analytical linear time invariant models from which the frequency response had to be calculated. In some practical applications, it may not be possible to describe a power system in terms of analytical models. However, what is available, and more importantly, what can be measured, is the frequency response. QFT requires only frequency response data to perform power system stabiliser design, thereby making it a powerful tool in this area.

Chapter 8

Practical Implementation

> Measure what is measurable, and make measurable what is not so.
>
> GALILEO GALILEI
> *The Armchair Science Reader*

The true benefit of any theoretical study is realised when application of that theory to real life situations is successful. Having designed and simulated a PSS in Chapter 7 for the power system laboratory, it is now appropriate to test the PSS design on the actual laboratory equipment.

This chapter documents the setup, testing, and results of that experiment. It also provides the link between theory and practice. For interest, the frequency response of the laboratory system was measured. A frequency range was chosen, and laboratory measurements are carried out at these points. The frequency range can be determined from knowledge of the practical power system characteristics, experience from operating the power system, or from a simple approximation of the power system dynamics. The aim of the exercise was to get a feel for the power system dynamics. The rest of the chapter deals with the results from the PSS tests. The PSS design from Case Study 2 is used.

Generally, a power system is stable if the interchange between electrical and mechanical energy is sufficiently damped. There are two concerns for damping. One is the "first swing", or transient and nonlinear stability. The other is linear mode damping. The power system stabiliser is typically designed for linear mode damping, and may possibly make the transient response worse. The power system stabiliser must improve the damping of the system in traditional approaches i.e. it must improve the damping in a robust way over the operating envelope. Robustness, as applied to the control system, is related to how the performance is affected by changes to the operating condition.

8.1 Frequency Response Measurements

From the bode plots of the laboratory power system model developed earlier, the frequency vector that was used is ω=`[0.1 1 3 5 10 12 15 20 25 30]` $\mathrm{rad\,s^{-1}}$. The relevant power system signals that are applicable to this experiment are the voltage reference input and the speed signal output. This choice of power system signals is consistent with most literature on power system stability [21].

A signal generator is used to superimpose a sinusoidal voltage waveform upon the voltage reference input to the AVR. The amplitude is chosen to be 5% of the steady state voltage reference signal. The reason for choosing 5% is so that the superimposed signal does not cause large deviations from the current operating point, and hence, does not cause the power system to operate nonlinearly.

The procedure for generating a frequency response data set is as follows:

1. set up the power system such that the AVR and governor loops are closed,

2. select the appropriate transmission line setting,

3. connect the synchronous generator to the infinite bus contactor via the transmission line. Ensure the contactor is open before making the connection,

4. connect the measuring instruments to the AVR voltage reference input and the speed transducer output,

5. switch on power to the prime mover, alternator, and infinite bus,

6. run the alternator up to synchronous speed and increase the alternator terminal voltage until it matches that of the infinite bus,

7. synchronise the alternator to the infinite bus only when the alternator voltages are in phase with the infinite bus voltages,

8. bring the alternator to the required operating condition by monitoring P_a and Q_a on the control panel meters. P_{base} is three times the armature power base value given in Appendix B. The wattmeter has a scale factor of 8. P_a in watts and Q_a in vars are calculated from the per unit power values P_b and Q_b as follows:

$$P_a = \frac{(P_b)(P_{base})}{(3)(8)} \tag{8.1}$$

$$Q_a = \frac{(Q_b)(P_{base})}{(\sqrt{3})(4)} \tag{8.2}$$

9. set the signal generator frequency to the required value and the amplitude to 5% of the voltage reference signal,

10. once the power system has attained steady state, the superimposed signal from the signal generator is switched on,

11. activate the data capture equipment to capture the superimposed input signal and speed transducer output signal of the power system,

12. save the data from the data capture equipment,

13. the superimposed signal frequency is changed to the next frequency in the frequency vector, and

14. steps 11 to 13 is repeated for all frequencies in the frequency vector.

Due to the large amount of noise on the speed transducer signal, it is impossible to measure the amplitude or phase of the signal by using the digital storage oscilloscope that is available in the laboratory. Therefore the data capture equipment was used and some post processing of the speed signal had to be done. It was decided that the speed signal would be captured along with the input signal. This data is then imported into Matlab where a fast fourier transform (FFT) is performed to extract the magnitude and phase of the two signals.

The program that performs this digital signal processing is found in Appendix F.17. Figure 8.1 shows an FFT plot of the speed data captured when the input signal to the AVR was modulated at 0.2Hz. Figure 8.2 shows the phase information. Alternatively, the cross power spectral density of the signals can be calculated, from which the magnitude and phase are obtained. Figure 8.3 shows the result of such an exercise for an input signal modulated at 0.2Hz. The maximum amplitude is at the 0.2Hz frequency bin. The displayed frequency has been limited to 0.5Hz for clarity. This technique is useful when the output signal is noisy, or when the output signal amplitude is too small to make sense in a FFT analysis.

The FFT is performed on all the data captured for the various frequencies. Table 8.1 summarises the magnitude and phase information for the speed signal and AVR input signal at all frequencies. When performing data capture for FFT analysis, some simple rules need to be followed to ensure that the magnitude and phase information can be extracted from the measured signal. They are

1. the sampling frequency determines the range of frequencies in the FFT,

2. the number of samples determine the frequency resolution of the FFT,

3. the phase information is relative to the start of the time domain signal,

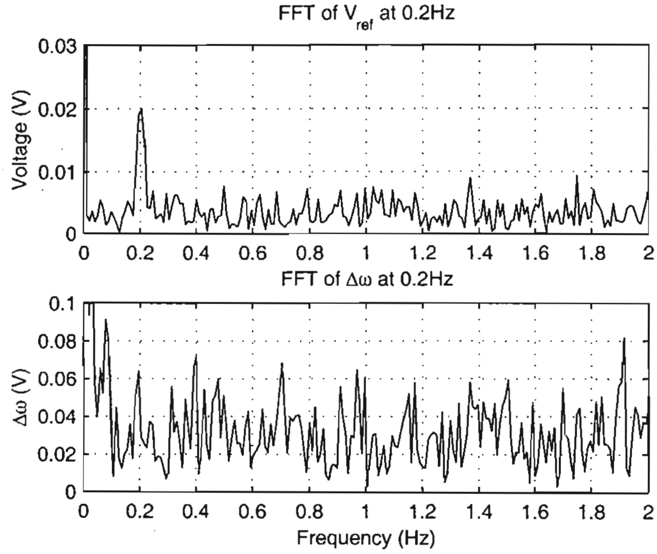


Figure 8.1: FFT magnitudes of V_{ref} (at 0.2Hz input frequency) and $\Delta\omega$.

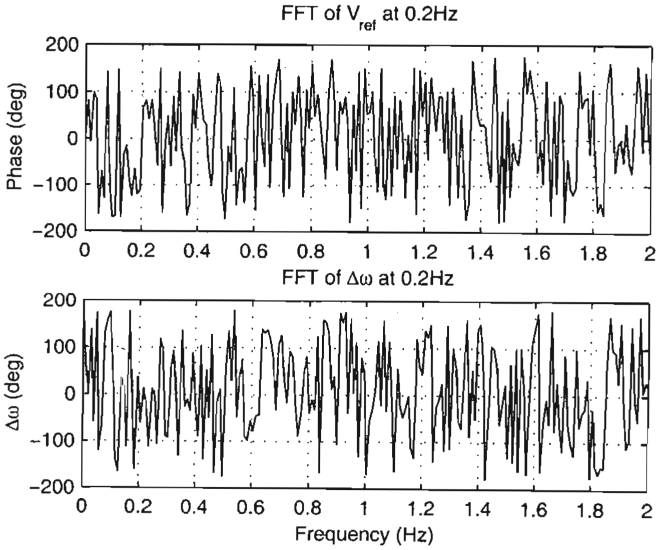


Figure 8.2: FFT phases of V_{ref} (at 0.2Hz input frequency) and $\Delta\omega$.

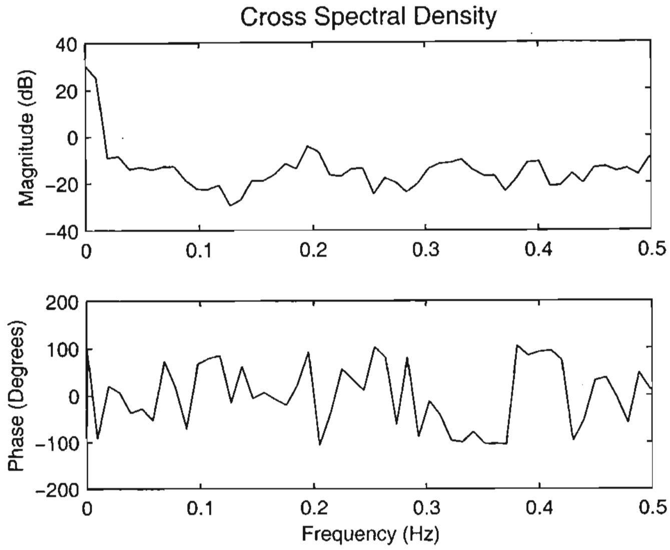


Figure 8.3: The cross power spectral density of the input and output signals.

Measured from FFT at P=100W, Q=173VAR, and V_t=128V				
Frequency(Hz)	$\|\Delta\omega\|$ (V)	$\|\Delta V_{ref}\|$ (V)	$\angle\Delta\omega$ (deg)	$\angle\Delta V_{ref}$ (deg)
0.2	56.6	14.3	35	−113
0.5	32.7	12.9	−32	−172
0.8	60.8	5.8	87	−149
1.6	3.7	25.5	115	115
2.0	28.8	58.3	166	−28
2.4	26.7	45.6	−107	57
3.2	21.7	29.4	−178	163
4.0	44.2	17.0	−96	−82.7

Table 8.1: The magnitude and phase of V_{ref} relative to $\Delta\omega$.

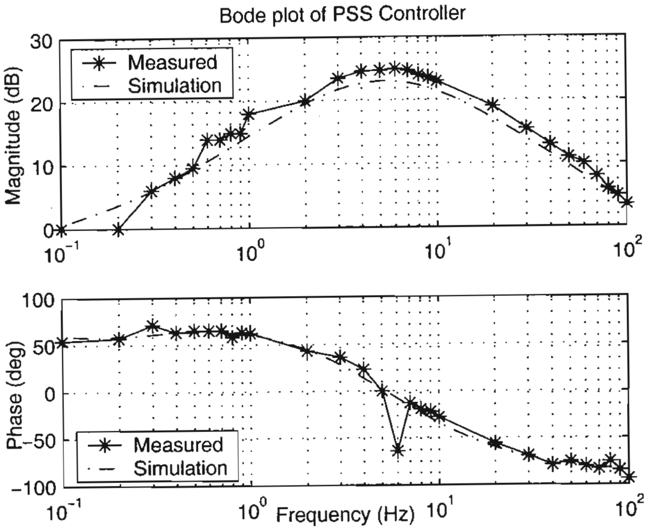


Figure 8.4: Bode plot of the designed PSS and simulation PSS.

4. the data capture must be triggered at the same point on the signal every time to obtain consistent results, and

5. to get the actual phase shift, subtract the reference signal phase from the output signal phase at the frequency of interest.

8.2 Measured PSS Frequency Response

The controller designed in Case Study 2 was tested by measuring its frequency response and comparing it to the simulation bode plot. This was done before the PSS was used in the experiment. It was to ensure that the controller was properly built, and that the required transfer function was realised. The input to the controller was swept over a frequency range of 0.01Hz to 100Hz in decades. The output signal magnitude and phase relative to the input signal were recorded. The results are tabulated in Appendix C.

A bode plot showing the actual frequency response, along with the predicted frequency response, is shown in Figure 8.4. The phase discrepancy of 50° at 6Hz is due to a measurement error. It can be seen that the controller has been correctly built and does in fact accurately synthesise the designed controller transfer function.

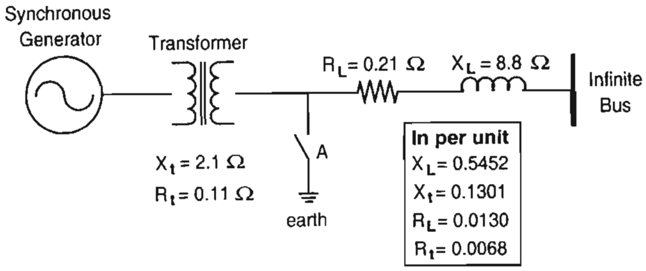


Figure 8.5: The power system setup.

8.3 PSS Tests and Results

A single line diagram showing the power system setup is shown in Figure 8.5. This is the same electrical circuit as used for the simulation in Case Study 2. Channels 0, 3, 4, and 6 on the data capture equipment were used to capture the speed signal, load angle, terminal voltage, and PSS output respectively. The short circuit at point (A) in Figure 8.5 was applied at different operating points. The same transmission line is used for all the tests.

8.3.1 Operating point 1

For this operating point, the active power P_a is set at 105 watts, the reactive power Q_a is set at 160 vars, and the terminal voltage V_t is set at 128 vac phase to neutral. From Equations 8.1 and 8.2, the per unit values are $P_b = 0.84$ p.u, $Q_b = 0.37$ p.u, and $V_t = 1.01$ p.u. First, the short circuit fault was applied without the PSS in circuit. Figures 8.6 to 8.8 show the result of the test. The noise in the measurements is probably due to the measurement setup, and the high frequency gain of the controller may have contributed significantly to the passing of noise to the measurements.

Then the test was performed with the PSS in the circuit. However, the PSS gain was set to 50% of the designed gain ($R_6 = 12\text{k}\Omega$). This was done from a safety viewpoint and to prevent equipment from being damaged should anything go wrong. The short circuit fault results with the PSS in the circuit are shown in Figures 8.9 to 8.11. Note that the short circuit duration is shorter than the open loop short circuit. This is due to the laboratory vacuum contactor being quite old and inaccurate with regard to timing.

It can be seen that the signals have high frequency components with large amplitudes which obscures the low frequency signals that are of interest. To view the low frequency signals, the data is passed through a low pass forward filter and a low pass backward filter in Matlab. The low pass filter

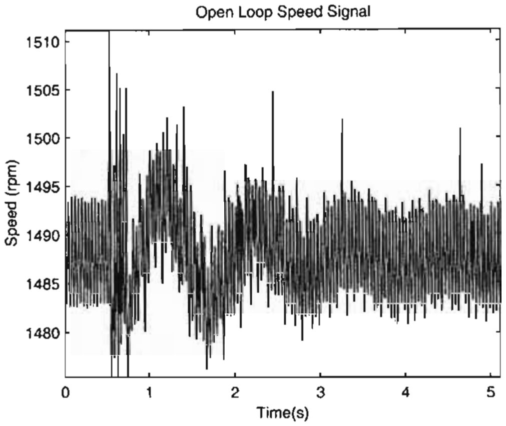


Figure 8.6: Speed signal without PSS.

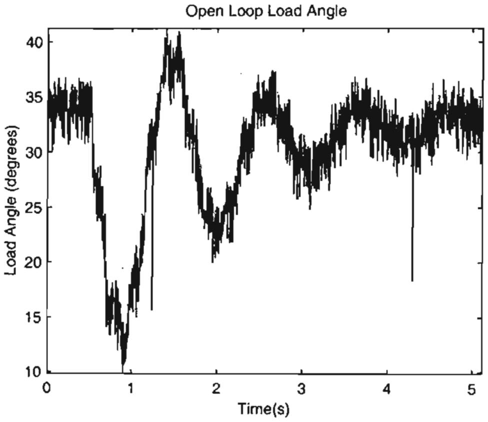


Figure 8.7: Load angle without PSS.

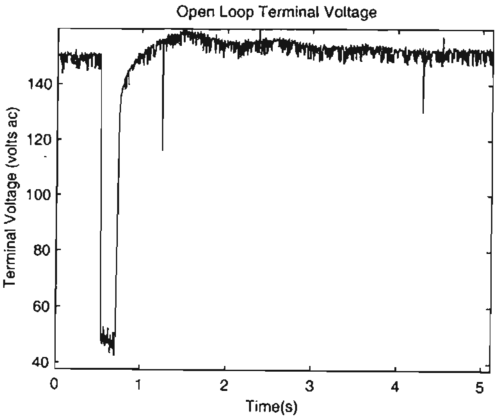


Figure 8.8: Terminal voltage without PSS.

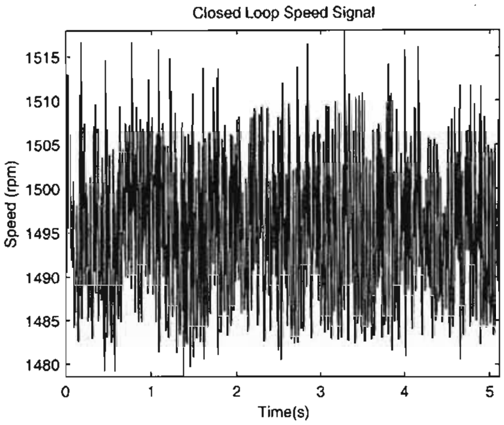


Figure 8.9: Speed signal with PSS at 50% gain.

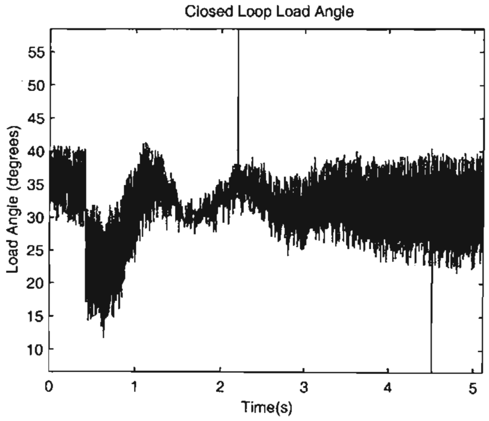


Figure 8.10: Load angle with PSS at 50% gain.

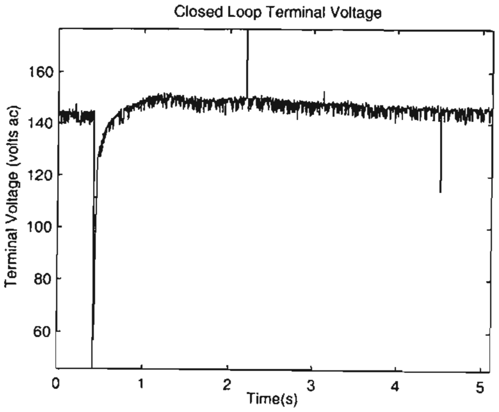


Figure 8.11: Terminal voltage with PSS at 50% gain.

is a third order Butterworth filter with a $-3dB$ frequency of $5\,\mathrm{rad\,s^{-1}}$.

The reason for using a forward and backward filter is so that the phase information can be preserved. The result of this filtering are shown in Figures 8.12 to 8.14. Measured data is generally not well behaved. Outliers (unusual data values) may occur as shown in Figure 8.12 and the controller may react to them. Comparisons of the measured signals in open and closed loop are also plotted. Figures 8.18 to 8.20 show the filtered signals with and without a PSS. The measured signals are filtered using the same low pass Butterworth filter discussed earlier.

The 5 rpm difference in Figure 8.18 could be due to the voltage supply in the laboratory or, more likely, to the speed measurement instrument. The theoretical value for the speed signal under steady state is 1500 rpm. The high frequency component of the measured signal is due to the high frequency gain of the controller passing noise to the measuring instrument. Note that in Figure 8.19, the PSS is affecting the first swing [21, p.867, Fig.E13.7], and not the damping.

Figure 8.20 shows the measured terminal voltage signal, and the effect of the PSS in closed loop. The gain of the PSS can affect the stability of the system quite drastically. For example, a $6dB$ reduction in PSS gain will result in stability problems if the power system was unstable in open loop, because of conditional stability. However, if the system was stable in open loop, then the reduction in PSS gain would have no effect.

The measured signals can also be used to verify the theory. By comparing the measured results to the simulation model results, the accuracy of the theoretical modelling can be confirmed. Figure 8.21 shows what the expected results should have been. The simulation was run under the same conditions as the laboratory power system. That is, same operating point, and same fault durations for the open and closed loop cases. From the comparison, it is apparent that the measurements do confirm the accuracy of the simulation results, and hence the theory.

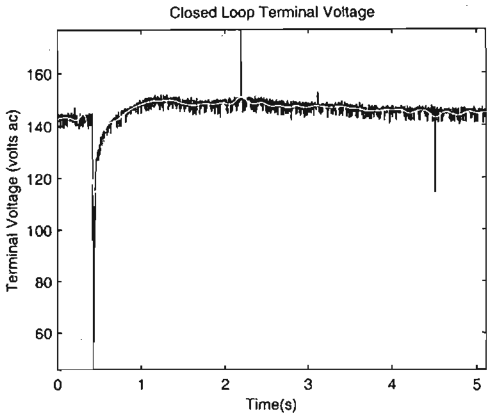


Figure 8.12: Filtered terminal voltage signal (white) and the original signal.

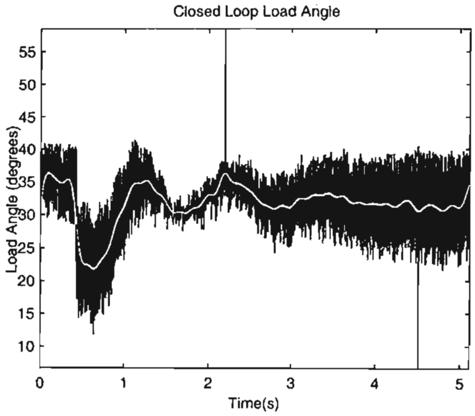


Figure 8.13: Filtered load angle signal (white) and the original signal.

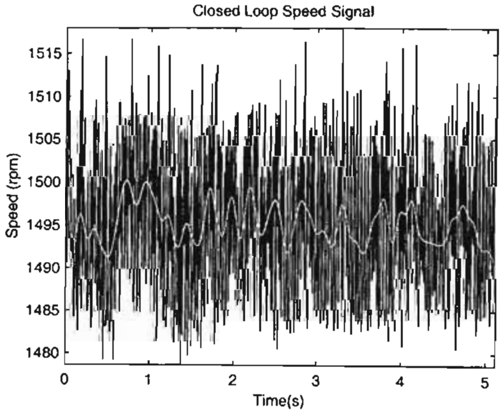


Figure 8.14: Filtered speed signal (white) and the original signal.

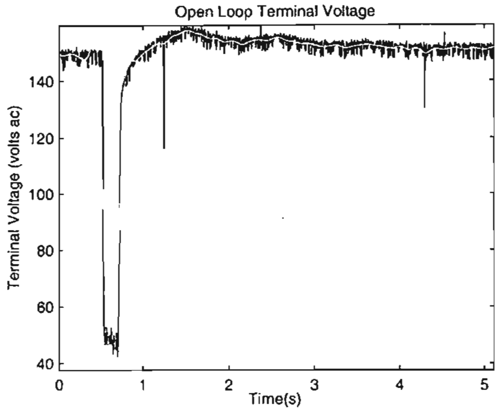


Figure 8.15: Filtered terminal voltage signal (white) and the original signal.

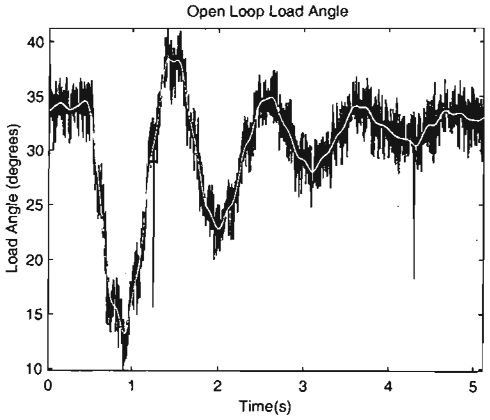


Figure 8.16: Filtered load angle signal (white) and the original signal.

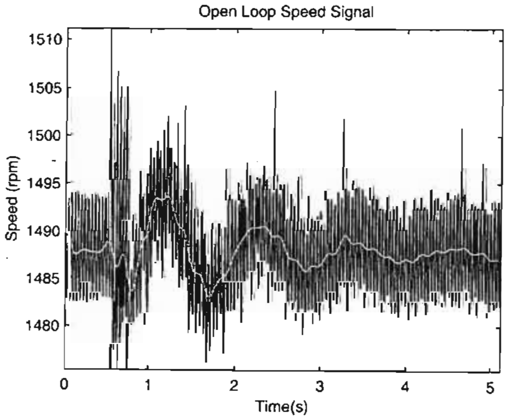


Figure 8.17: Filtered speed signal (white) and the original signal.

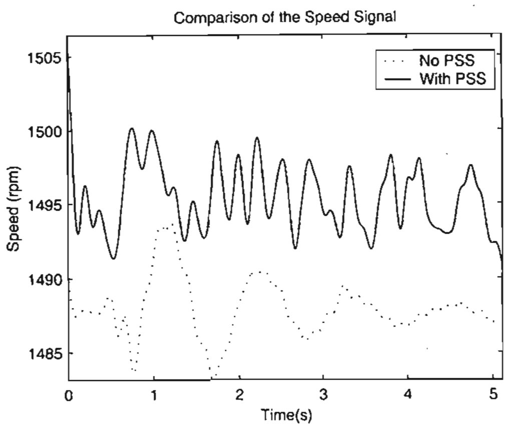


Figure 8.18: Speed signal without and with a PSS.

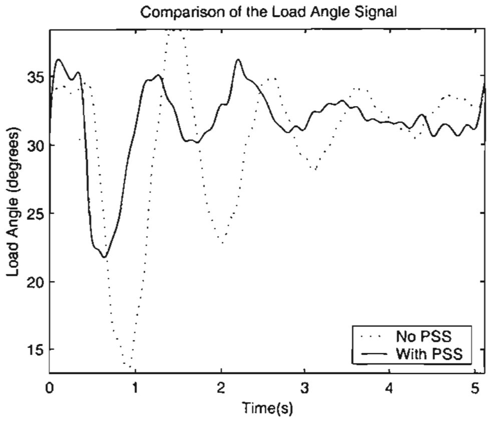


Figure 8.19: Load angle signal without and with a PSS.

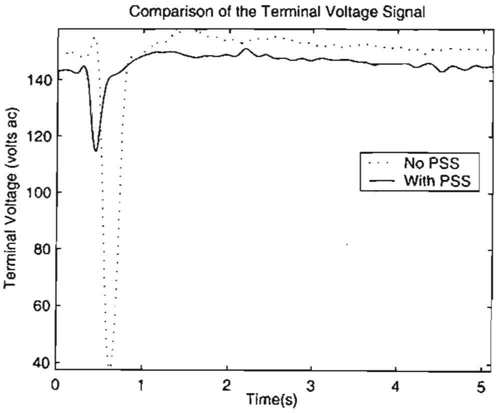


Figure 8.20: Terminal voltage signal without and with a PSS.

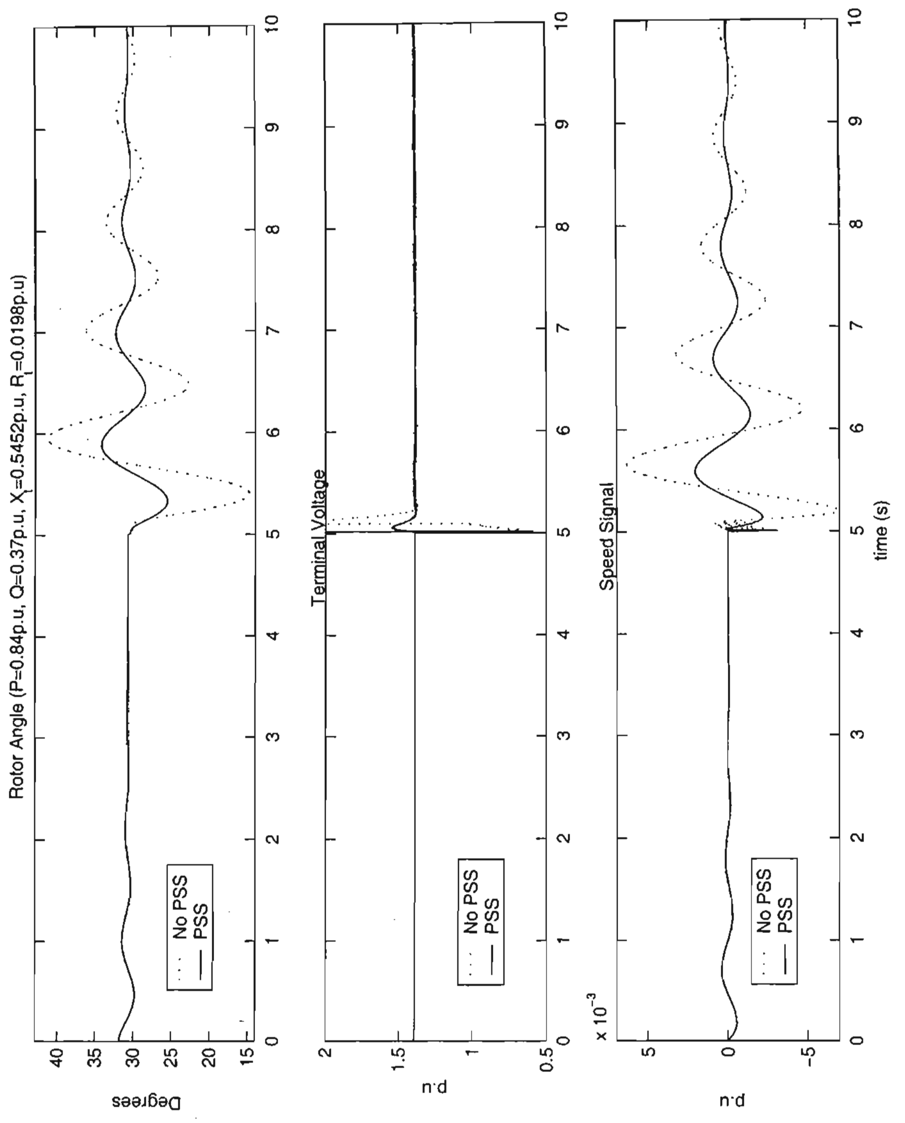


Figure 8.21: Simulation results at 50% gain.

8.3.2 Operating point 2

This was a lightly loaded operating point with $P_a = 20$W, $Q_a = 60$VAR, and a terminal voltage of 138V phase to neutral. From Equations 8.1 and 8.2, the per unit values are $P_b = 0.16$ p.u, $Q_b = 0.13$ p.u, and $V_t = 1.09$ p.u. The short circuit fault was applied without the PSS in the circuit. The result is shown in Figure 8.22, with the filtered signals shown in white. Then the PSS loop was closed with a PSS gain of 50%. The same fault is applied, and the captured data is shown in Figures 8.23, along with the low pass filtered signals. The compensated and uncompensated systems are shown in Figure 8.24.

A simulation was carried out at this operating point, to test the accuracy of the theory. Figure 8.25 shows that the load angle phase difference between the compensated and uncompensated system matches that of the measured system. The magnitude of the simulated load angle is also similar to the measured load angle.

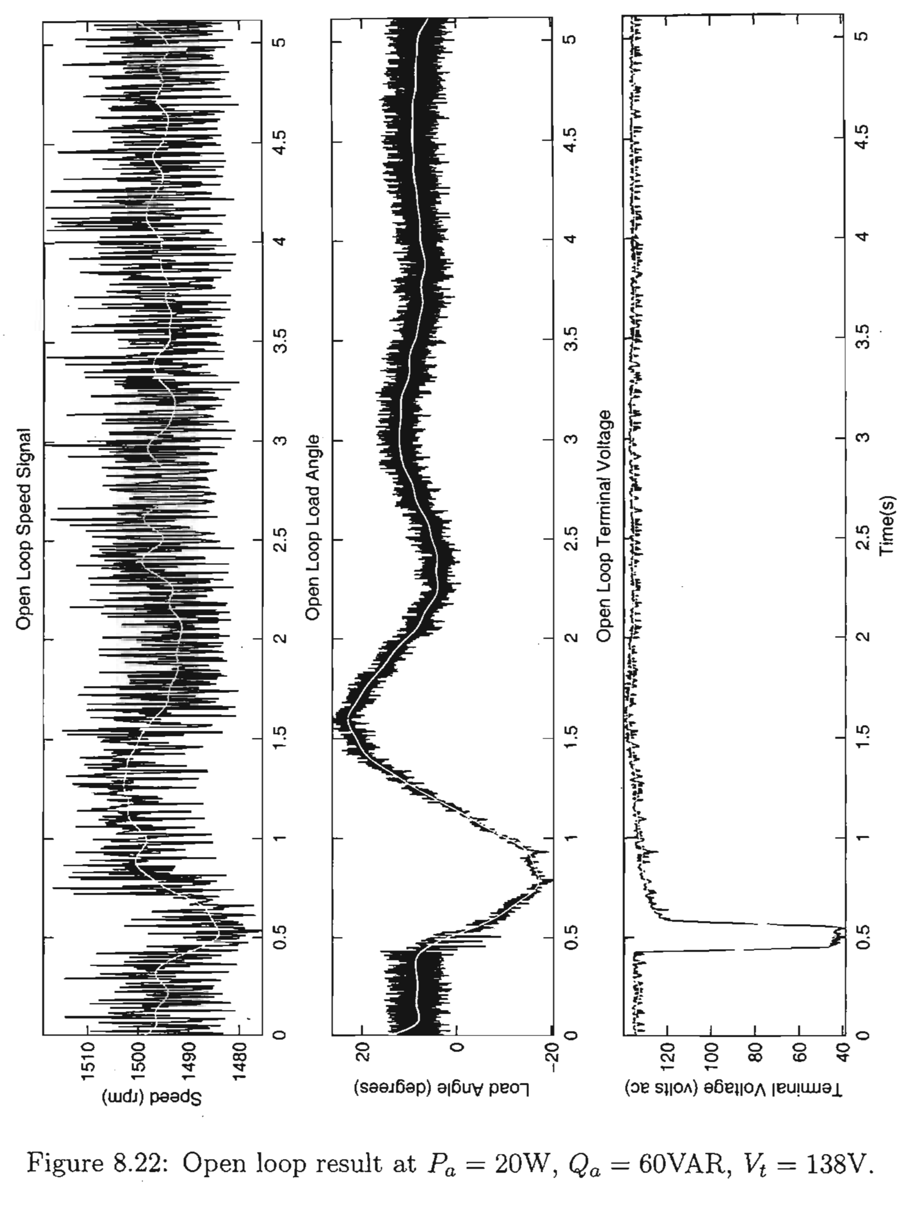


Figure 8.22: Open loop result at $P_a = 20\text{W}$, $Q_a = 60\text{VAR}$, $V_t = 138\text{V}$.

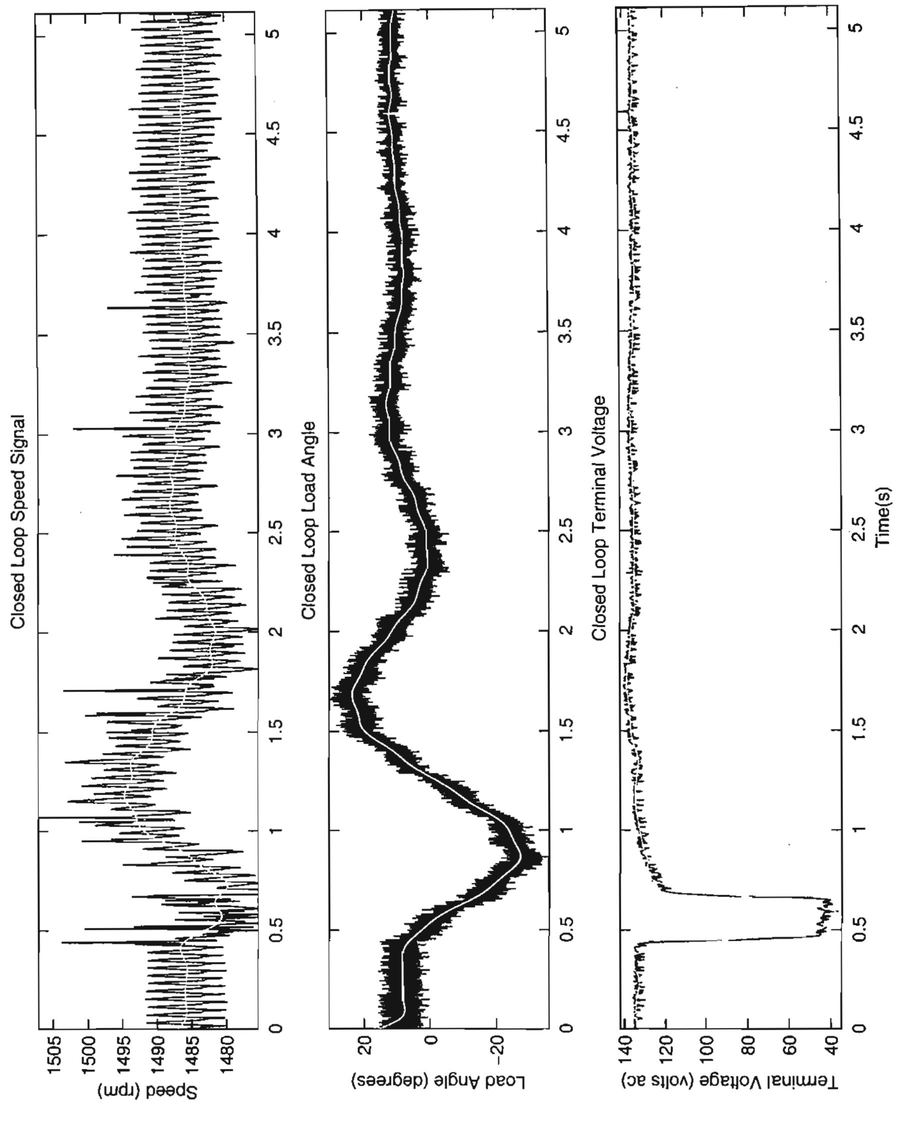


Figure 8.23: Closed loop result at $P_a = 20$W, $Q_a = 60$VAR, $V_t = 138$V.

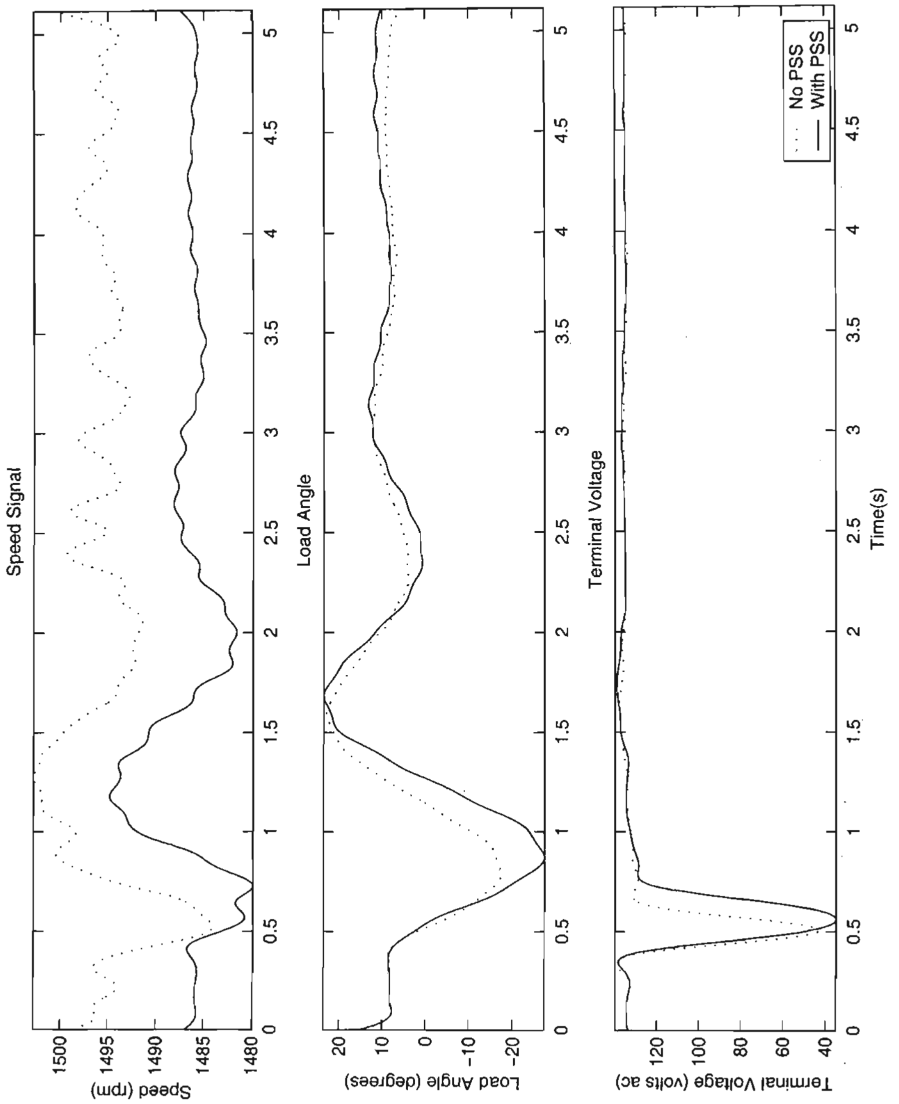


Figure 8.24: Comparison of results at $P_a = 20$W, $Q_a = 60$VAR, $V_t = 138$V.

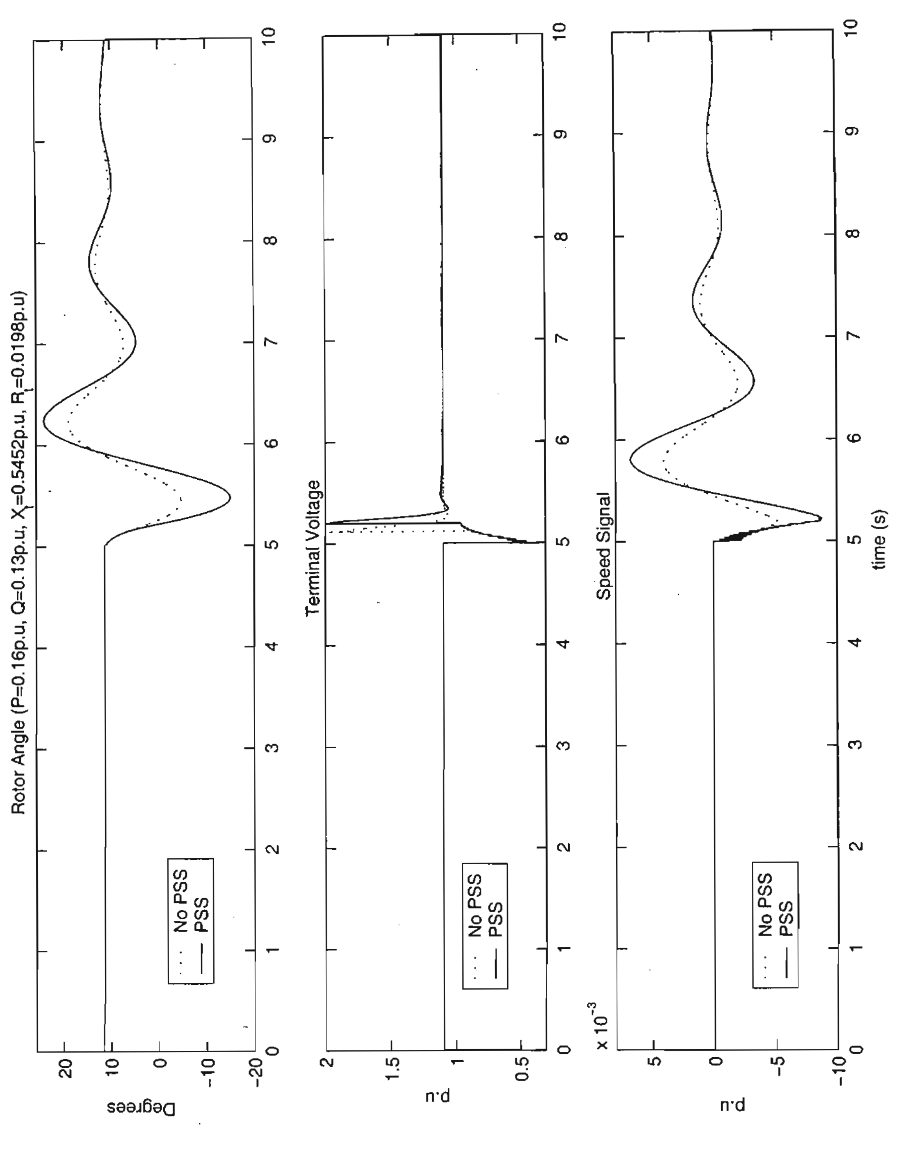


Figure 8.25: Operating point 2 simulation results at 50% gain.

8.3.3 Operating point 3

For this operating point, the active power was set at 105 watts, the reactive power at 160 vars, and the terminal voltage at 146 vac phase to neutral. The per unit values from Equations 8.1 and 8.2 are $P_b = 0.84$ p.u, $Q_b = 0.37$ p.u, and $V_t = 1.15$ p.u. This time, the PSS gain was set to 100% ($R_6 = 25\text{k}\Omega$). The short circuit fault was applied without the PSS in the circuit and the result is shown in Figure 8.26. The fault was then applied with the PSS loop closed. The result is shown in Figure 8.27. The plots in white are the signals with the low pass filter applied. Increasing the PSS gain has increased the damping of the power system.

The simulation results are shown in Figure 8.29. By comparing the phase and magnitude of the measured load angle to the simulated load angle, it can be seen that these results are similar, once again validating the theoretical model.

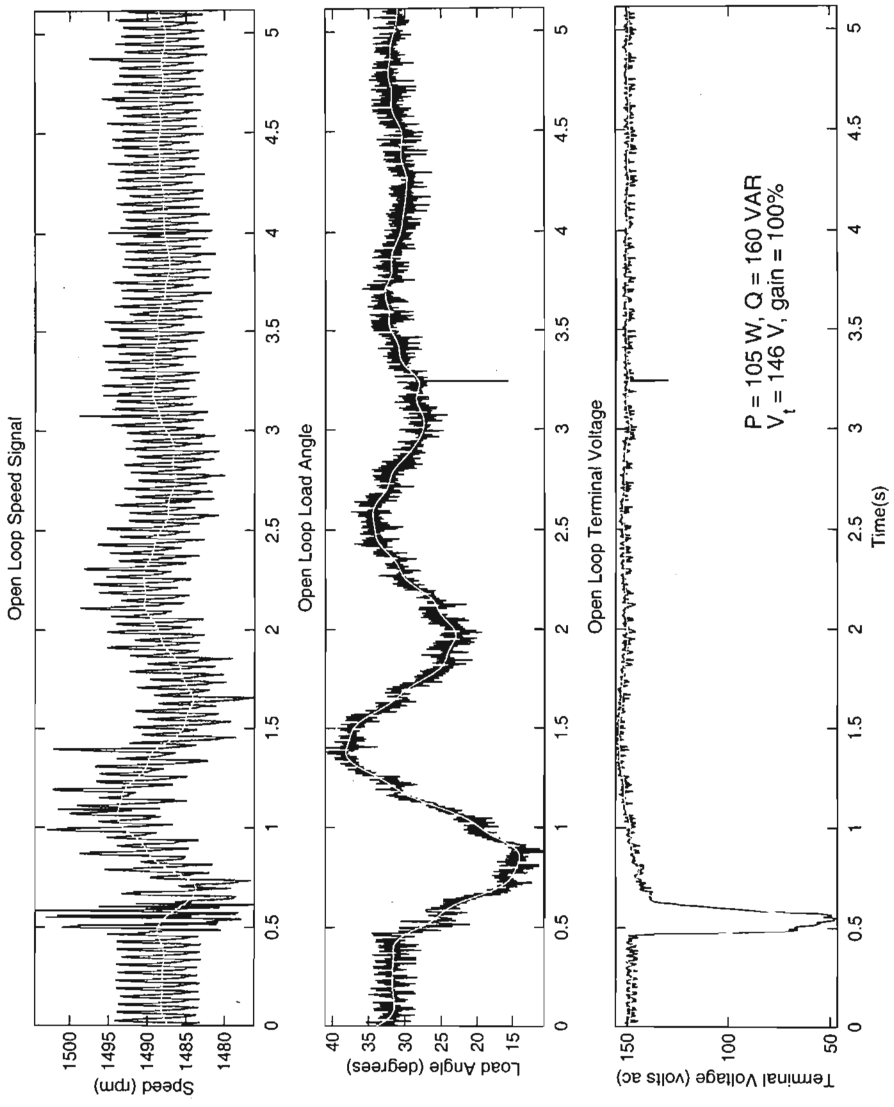


Figure 8.26: Open loop result at $P_a = 105\text{W}$, $Q_a = 160\text{VAR}$, $V_t = 146\text{V}$.

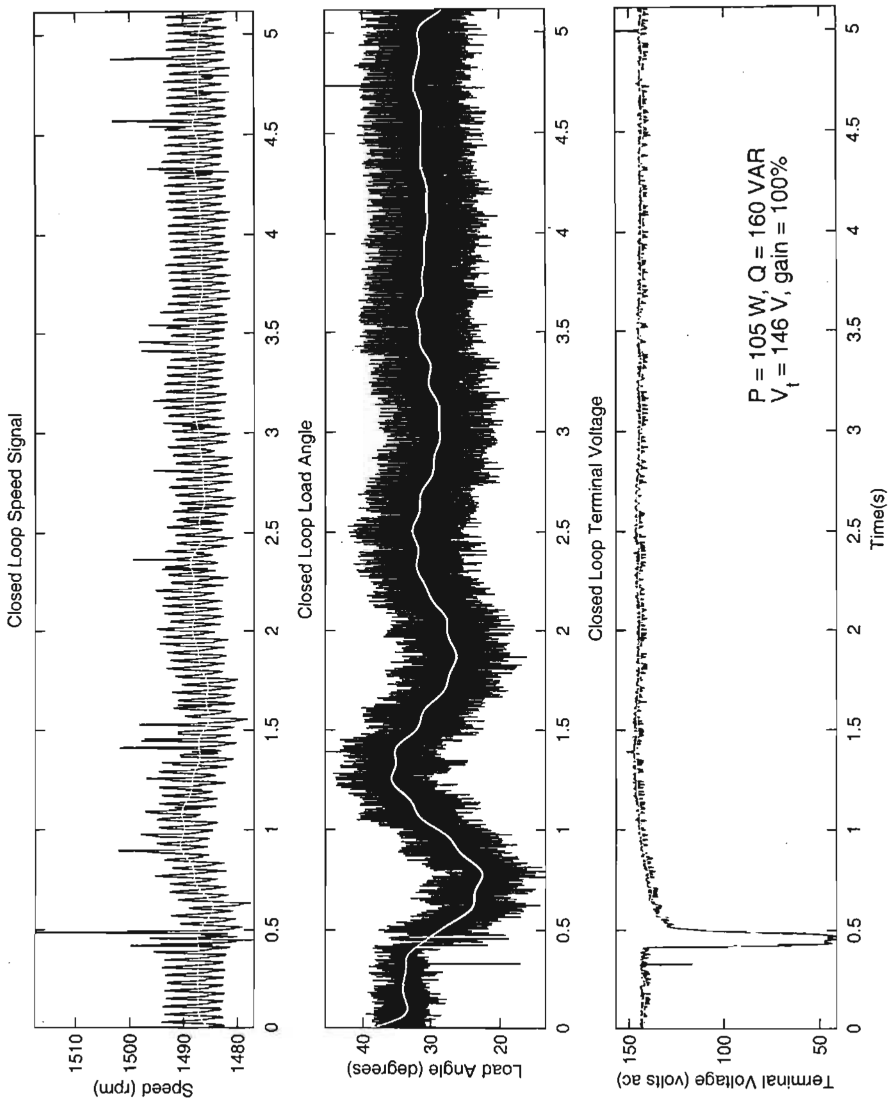


Figure 8.27: Closed loop result at $P_a = 105$W, $Q_a = 160$VAR, $V_t = 146$V.

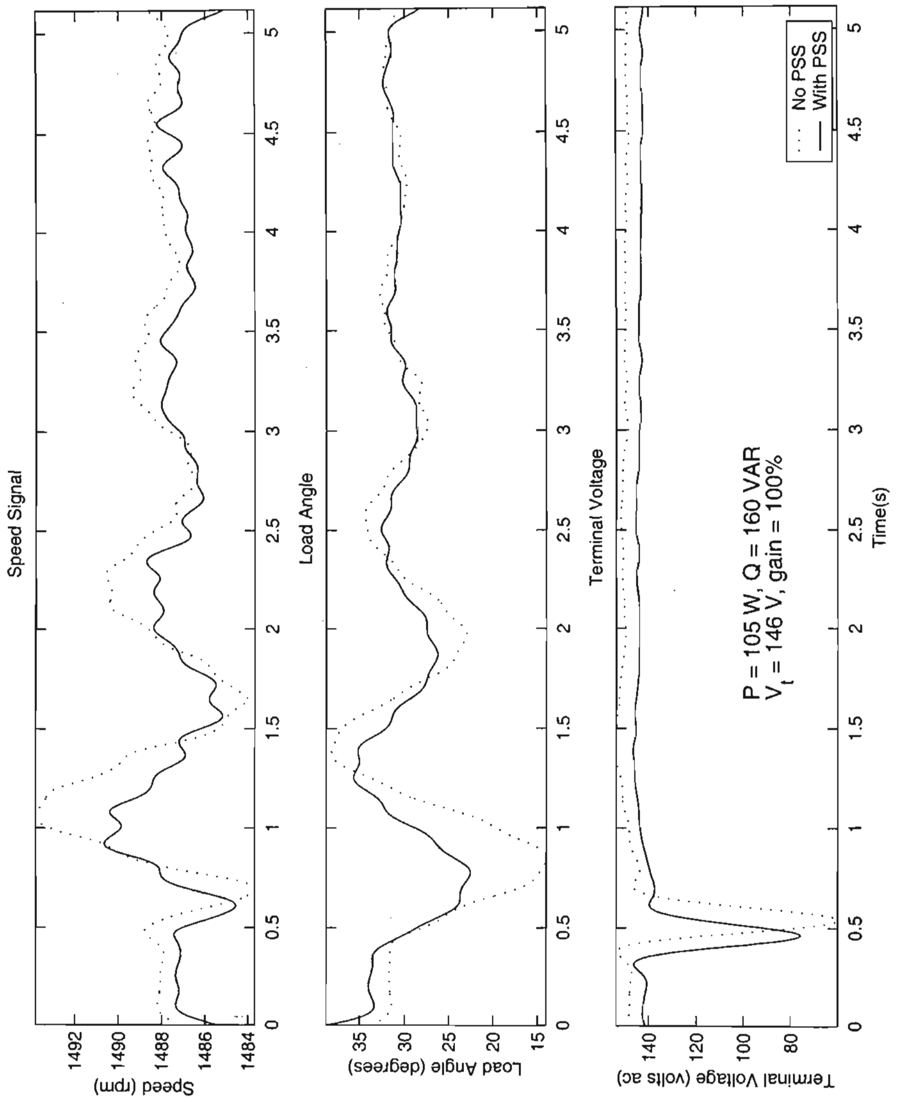


Figure 8.28: Comparison of results at P_a = 105W, Q_a = 160VAR, V_t = 146V.

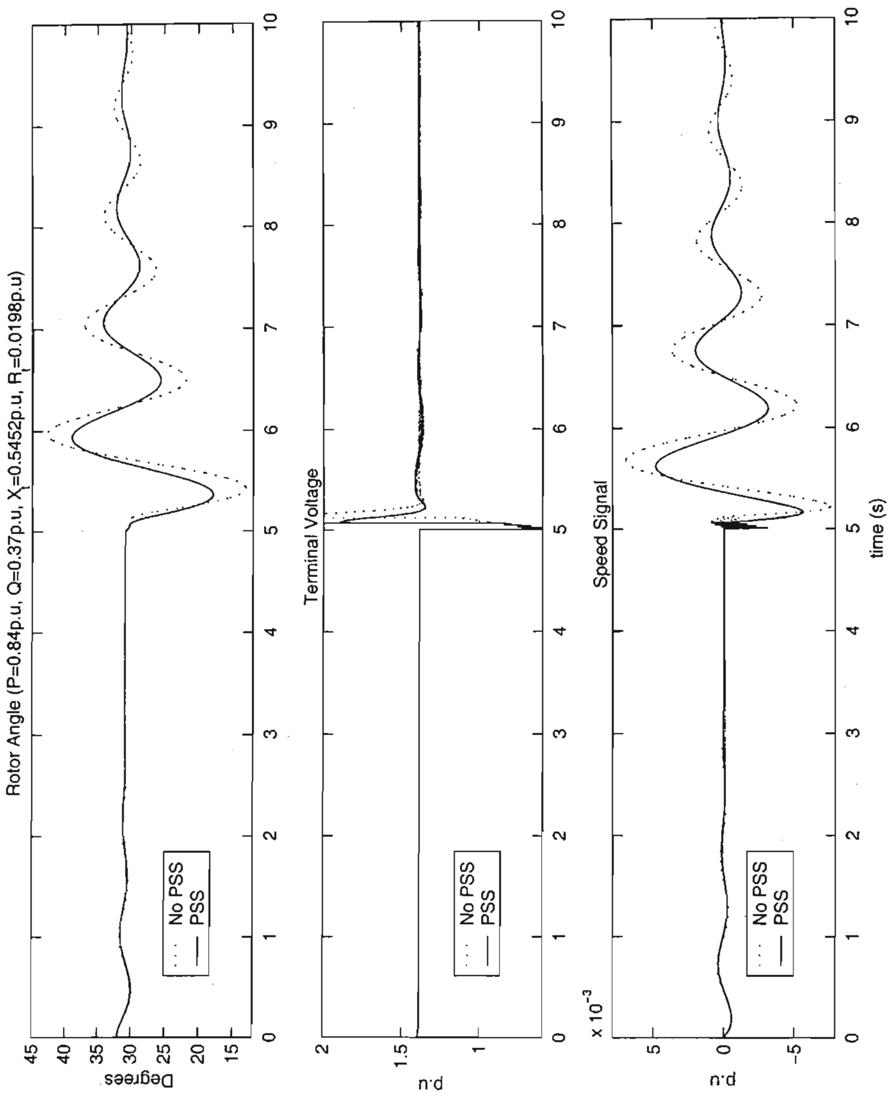


Figure 8.29: Operating point 3 simulation results at 100% gain.

8.3.4 Operating point 4

This was a lightly loaded operating point with P_a = 20W (0.16 p.u), Q_a = 60VAR (0.13 p.u) and V_t = 132V (1.04 p.u) phase to neutral. The PSS gain was set at 100% ($R_6 = 25\text{k}\Omega$). First, the fault was applied without the PSS in the circuit. The result is shown in Figure 8.30. Then the PSS loop was closed and the fault applied. Figure 8.31 shows the result. The white plots once again represent the signals after they have been filtered. Figure 8.32 shows that the PSS provides little or no damping for this light loading condition.

Simulation results were obtained for this operating point and they are shown in Figure 8.33. However, there is a discrepancy between the theoretical and measured results. The fault duration for the open loop simulation was 0.035 seconds, and the closed loop duration fault was 0.25 seconds. These values were obtained from the plots of the measured terminal voltage for the open and closed loop cases. The reason for the different fault durations is due to the vacuum contactor not opening, or closing properly. The simulations were run using the measured fault durations for the open and closed loops cases.

The conclusion from this comparison is that the measurements and simulations do not match. Future work that emphasises the practical implementation would have to repeat this test.

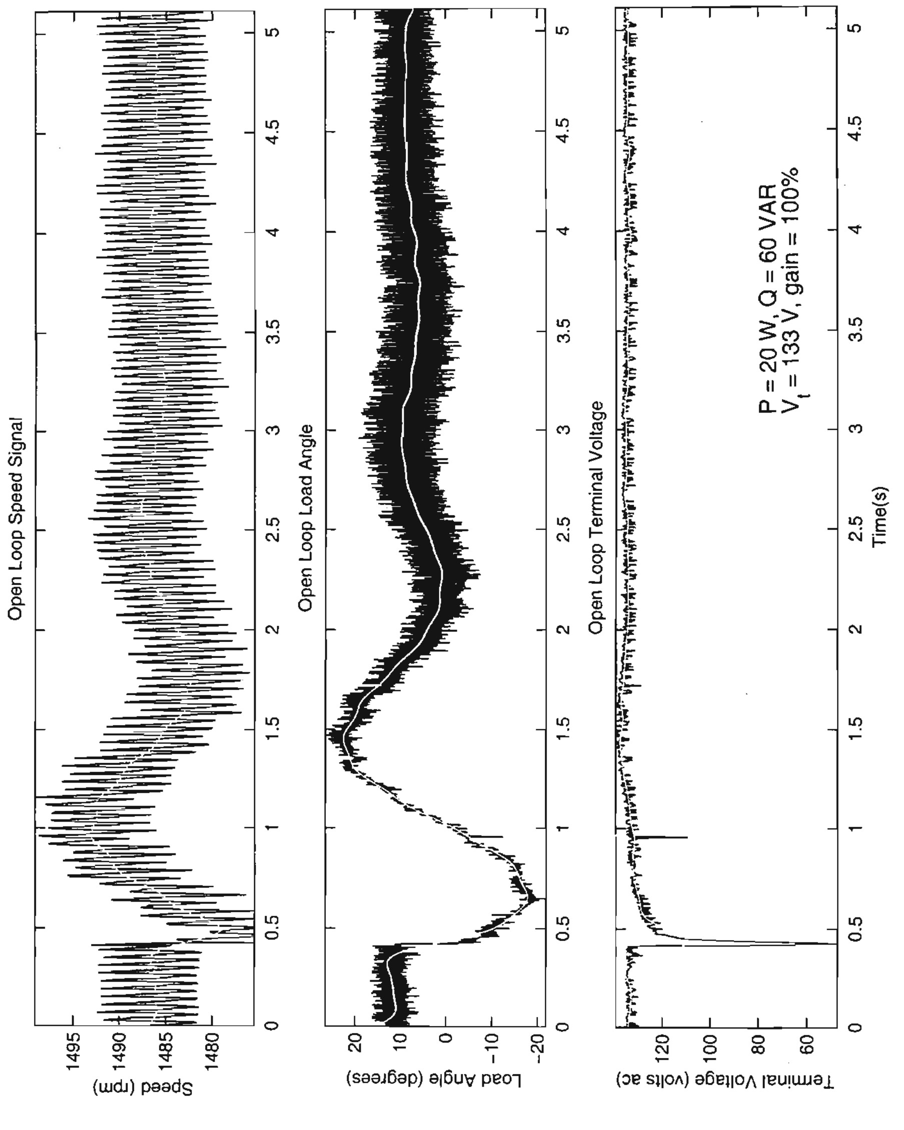


Figure 8.30: Open loop result at $P_a = 20\text{W}$, $Q_a = 60\text{VAR}$, $V_t = 132\text{V}$.

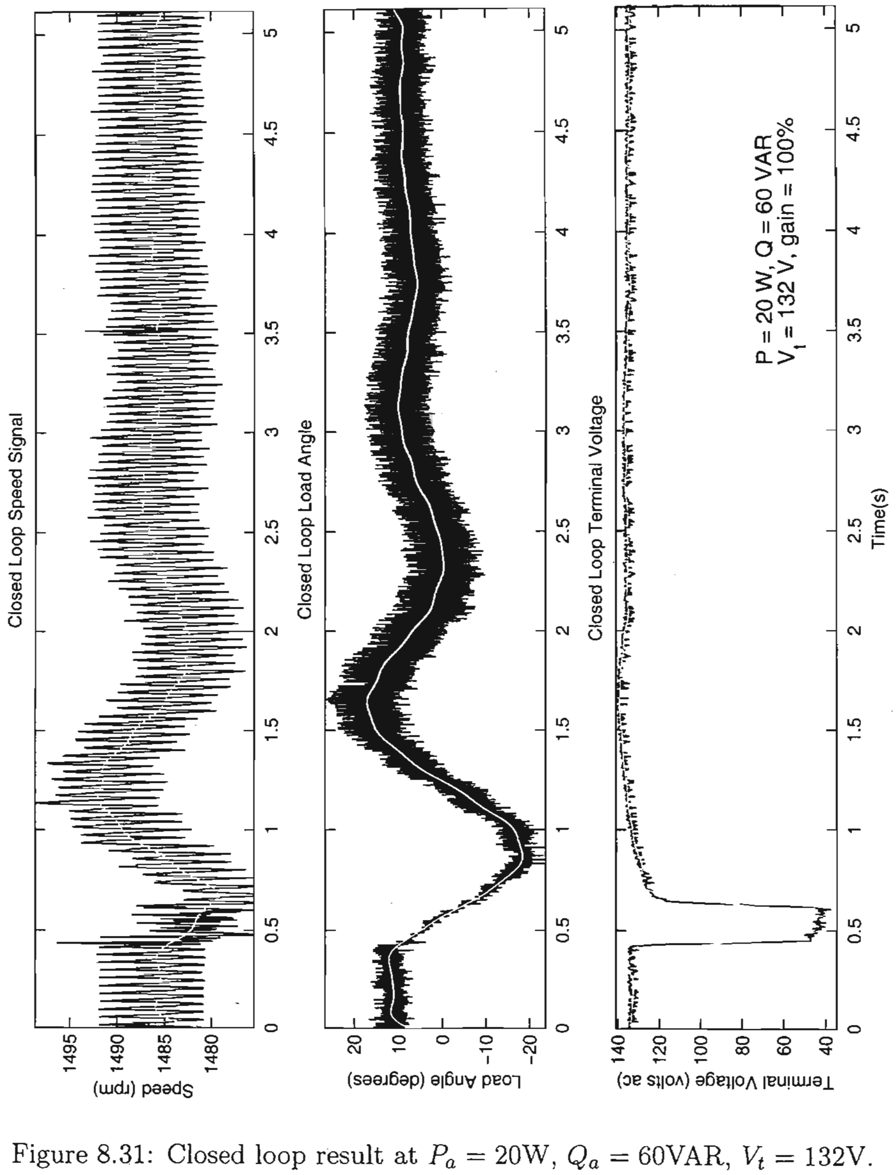


Figure 8.31: Closed loop result at $P_a = 20\text{W}$, $Q_a = 60\text{VAR}$, $V_t = 132\text{V}$.

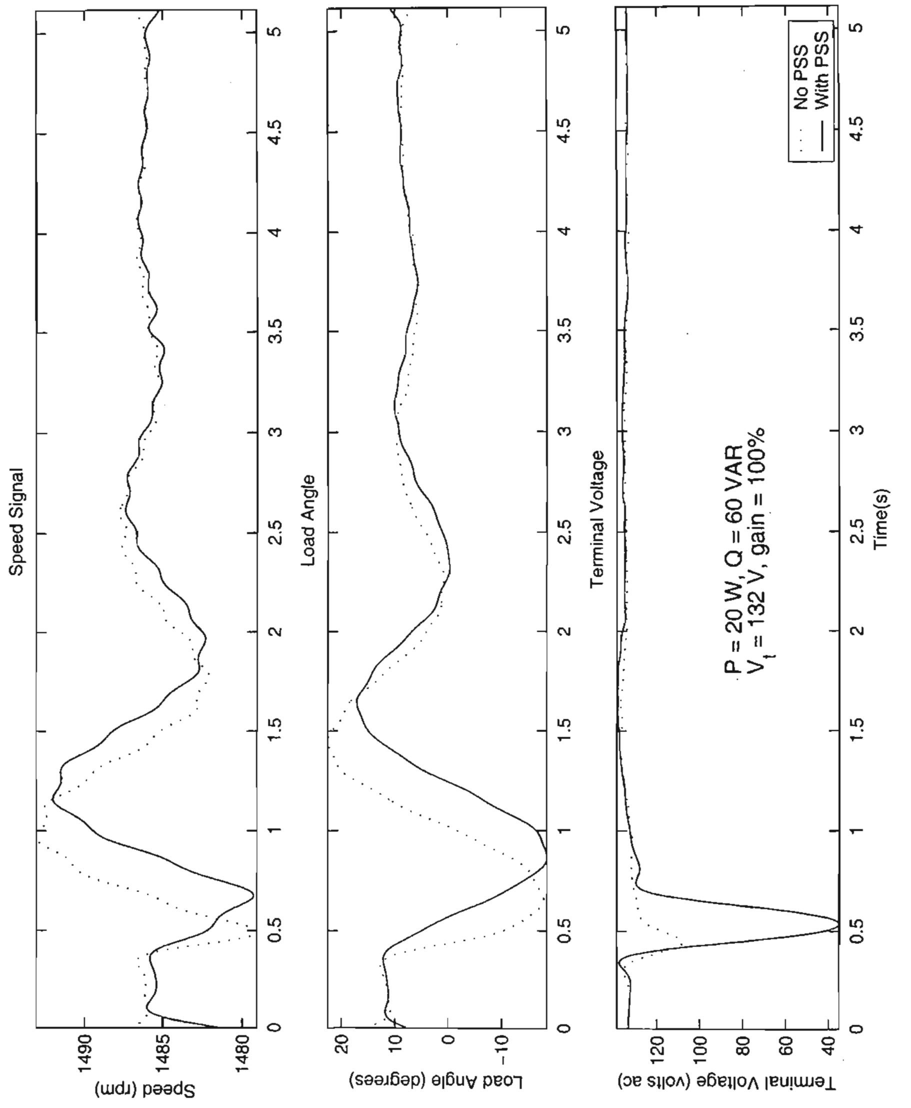


Figure 8.32: Comparison of results at $P_a = 20$W, $Q_a = 60$VAR, $V_t = 132$V.

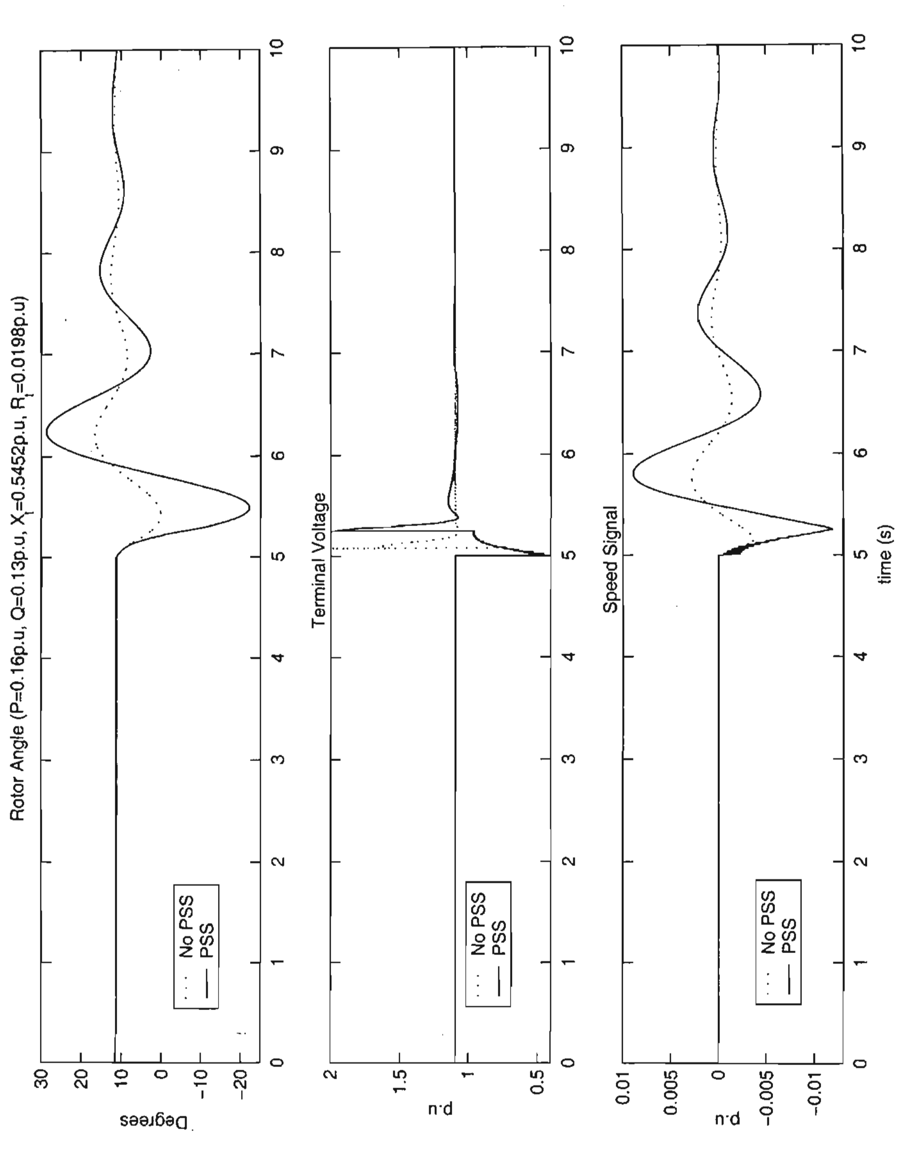


Figure 8.33: Operating point 4 simulation results at 100% gain.

8.4 Summary

The results in this chapter have verified the use of QFT as a powerful tool in the design of power system stabilisers. Due to the inherent high damping of the power system laboratory, the performance criteria could not be met. To achieve the performance specification, gain greater than $20dB$ was required. From Figure 8.4, $20dB$ dc gain is equivalent to $43dB$ gain at 6Hz. This high gain is not feasible as signals higher than 100mV would cause the output control voltage to swing to the operational amplifier rail voltage. Since the PSS supply voltage is 12V, the 2V drop for a 741 operational amplifier gives a rail voltage of 10V. This would result in saturation of the power system stabiliser.

QFT is a frequency domain design tool, and it is preferable to work with measured frequency data. However, not all physical systems allow for frequency response measurements, and the laboratory power system is one of them. The difficulty of obtaining frequency response data became apparent when the speed signal measurements were made. The extremely noisy and small signal necessitated the use of digital signal processing, and techniques such as FFT and spectrum analysis could not yield a meaningful result.

The QFT design process did ensure robust stability at all design operating conditions. This was proved in the laboratory by applying a fault to earth on the power system at different operating points. In each case, the generator maintained synchronism and returned to its pre-fault steady state condition after application of the fault.

Chapter 9

Conclusions

> I hope that posterity will judge me kindly, not only as to the things which I have explained, but also for those which I have intentionally omitted so as to leave others the pleasure of discovery.
>
> RENÉ DESCARTES
> *La Geometrie*

QFT has a wide variety of applications in different fields. Power system stabilisation has been a topic under consideration for quite some time now. The focus of using QFT to solve power system stability problems has occurred quite recently.

The merging of these engineering design methods has given a remarkable insight on how to produce results that satisfies the client's expectations, yet still uses sound engineering principles. There are other control design methods such as H_∞, model-predictive control, and root locus, that can be applied to power system stabiliser design, but their solutions are either more complex, or only solves the problem at a particular operating point.

Chapter 2 has shown the pioneering work carried out to produce stabilisers that improves system performance. More importantly, the stability phenomenon was identified, and research done to find an simple and robust solution to the power system stability issue. All the design methods required an analytical model of the power system. Generally, power systems are difficult to analyse and the models are quite complex. Often, simplified models are used, but this simplification, unless done wisely, leads to a loss of accuracy in terms of representing the power system as a whole.

The increase in computer processing power means that nonlinear simulation of high order systems has become more practical. An understanding of the underlying theory of power systems is still essential, and Chapter 3 provided the background necessary to understand how power systems work and how they are modelled. It was shown that the equations that describe the components of a power system can be tedious and difficult to solve without a computational tool.

Chapter 4 introduced the emerging technology of computer aided design to solve complex engineering problems. Using CAD technology, and understanding how the CAD system models a particular piece of equipment, is paramount to generating accurate simulations and predicting real-life behaviour of the power system. Once the model of the power system was available for analysis and observation, Chapter 5 introduced the tool that would enable the power system stability problem to be solved.

The basics of QFT was discussed, as well as its historical development. Reasons for why QFT has evolved into what it is today were offered. The uncertainty concept was introduced and examples were given that showed how this frequency domain technique used uncertainty to design robust controllers. A CAD environment offered the benefits of ease of use and reduction of manual computational effort. This freed the designer to concentrate on engineering design rather than mathematical calculations. QFT is an iterative process, and the QFT toolbox allows a number of designs to be done quickly, each design hopefully improving on the previous one. QFT is a practical control design tool, and Chapter 6 introduced the vehicle on which the designed power system stabiliser would be tested.

The laboratory power system was designed to accurately represent an industrial power system many times its rating. It has the necessary power system components such as a steam turbine prime mover simulator, a distributed mass shaft to account for shaft twist in real applications, and a synchronous generator whose parameters in per unit are equal to a full sized machine. All the mechanical and electrical parameters for the laboratory electrical machinery are available, allowing accurate simulation of the physical system before practical work is carried out. Due to concerns about machine stability and equipment lifespan in the early days (1970), when the laboratory was first put together, the power system controllers (AVR and governor) designs are quite conservative. This resulted in a power system setup that is over damped.

However, Chapter 7 took the results of a practical test on the laboratory system and compared it to a Matlab simulation of the laboratory system. The matching results gave the confidence necessary to trust the simulation model. The nonlinear simulation model was converted to a linear model suitable for QFT design. Chapter 6 contained two case studies that used the laboratory simulation models. The first Case Study used a first order AVR model instead of the conservative laboratory AVR model. Its purpose was twofold. Firstly to show the laboratory power system response if the existing AVR circuit had to be changed. Secondly, to perform QFT design on a lightly damped system.

Case Study 2 used the laboratory AVR model. The aim of this case study was to show how QFT is used to design a power system stabiliser for an existing system, and how to implement the controller.

Chapter 8 made the link between theory and practice. The effectiveness and performance of the designed power system stabiliser is tested in the laboratory under different conditions. The results show that QFT design of power system stabilisers results in a controller that is simple to implement. It also proves that power system stabilisers can be designed using QFT, and that the designs do work.

Further work should be done to test the robustness of the PSS in the laboratory. The test at operating point 4 in the previous chapter will have to be carried out again. The method of obtaining the frequency response data from the nonlinear PSB model, instead of measurement in the laboratory, should also be investigated.

The conclusion is that QFT is a most suitable tool for power system stabiliser design.

Bibliography

[1] B Adkins and RG Harley. *The General Theory of Alternating Current Machines.* John Wiley and Sons, 1975. 20, 56, 150, 152

[2] SI Ahson and BW Hogg. Application of Multivariable Frequency Response Methods to Control of Turbogenerators. *International Journal of Control*, 30(4):533–548, 1979. 5

[3] JC Balda. *Subsynchronous Resonance of Turbogenerators with Particular Reference to a Few Selected Countermeasures.* PhD thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, November 1985. 56

[4] RC Beck. The Analysis of AC Machines in Power Systems. Master's thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, November 1986. 56, 155

[5] H Bode. *Network Analysis and Feedback Amplifier Design.* Van Nostrand, New York, 1945. 37

[6] E Boje and G Jennings. Quantitative Feedback Design of Robust Power System Stabilisers for Single Machine Infinite Bus Problems. Submitted for publication, 1998. 13, 16

[7] E Boje, O Nwokah, and G Jennings. Quantitative Design of SMIB Power System Stabilisers using Decoupling Theory. *14th Triennial World Congress, Beijing, P.R. China, IFAC*, pages 267–272, 1999. 1, 14, 101

[8] I Boldea and SA Nasar. *Electric Machine Dynamics.* Macmillan Publishing Company, 1986. 22

[9] C Borghesani, Y Chait, and O Yaniv. *Quantitative Feedback Theory Control Toolbox.* Terasoft, March 2001. 38, 40, 41, 42, 44, 46, 49, 53, 88

[10] FP DeMello and C Concordia. Concepts of Synchronous Machine Stability as Affected by Excitation Control. *IEEE Transactions on Power Apparatus and Systems*, PAS-88(4):316–329, April 1969. 3, 16

[11] E Eitelberg. Quantitative Feedback Design for Tracking Error Tolerance. *Automatica*, 36:319–326, 2000.

[12] Z Fadlalmoula, SS Robertson, J Reilly'O, and WE Leithead. Individual Channel Analysis of the Turbogenerator with a Power System Stabiliser. *International Journal Of Control*, 69(2):175–202, 1998. 10, 14, 16

[13] RE Fairbairn. *Turbogenerator Analysis with Special Reference to Subsynchronous Resonance and Parameter Estimation.* PhD thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, December 1991. 56, 155

[14] I Horowitz. Survey of Quantitative Feedback Theory. *International Journal of Control*, 53(2):255–291, 1991. 16, 36, 37

[15] I Horowitz and M Sidi. Synthesis of Feedback Systems with Large Plant Ignorance for Prescribed Time-Domain Tolerances. *International Journal of Control*, 16(2):287–309, 1972. 36, 37, 42

[16] FM Hughes and AMA Hamdan. Design of Turboalternator Excitation Controllers Using Multivariable Frequency-Response Methods. *Proc. IEE*, 123(9):901–905, September 1976. 5

[17] Hydro-Québec Research Institute. *Power System Blockset User Guide.* The MathWorks. Inc., 24 Prime Park Way, Natick, MA, 1.1 edition, January 1999. 28, 29, 31

[18] GD Jennings. *Modelling and Analysis of Turbogenerators in Single Machine and Multimachine Subsynchronous Resonance Studies.* PhD thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, October 1987. 56

[19] M Kelemen, AF Okou, O Akhrif, and LA Dessaint. Robust Control of a Synchronous Power Generator:Good Performance, Simple Controllers. Electrical Engineering Department, École de Technologie Supérieure, Montréal, Québec, Canada. 1, 2

[20] Paul C Krause. *Analysis of Electric Machinery.* McGraw-Hill Inc., 1986. 20, 23, 25, 168

[21] P Kundur. *Power System Stability and Control.* McGraw-Hill Inc., 1994. 2, 20, 79, 114, 122, 150, 158, 160

[22] MA Lahoud. *The Analysis and Control of Synchronous and Asynchronous Machines in Power Systems.* PhD thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, October 1982. 56, 57, 58

[23] K Lakmeerharan and M.L Coker. Optimal Placement and Tuning of Power System Stabilisers. *IEEE Transactions on Power Apparatus and Systems*, IEEE, SAPSSI, 1999. ESKOM Technology Group, Private Bag 40175, Cleveland, 2022, South Africa. 5, 14

[24] EV Larsen and DA Swann. Applying Power System Stabilisers Part I: General Concepts. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(6):3017–3024, June 1981. 2, 5, 8

[25] EV Larsen and DA Swann. Applying Power System Stabilisers Part II: Performance Objectives and Tuning Concepts. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(6):3025–3033, June 1981. 7, 8

[26] EV Larsen and DA Swann. Applying Power System Stabilisers Part III: Practical Considerations. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(6):3034–3046, June 1981. 8, 16

[27] DJN Limbeer. *Analysis and Control of AC Rotating Machines in the Presence of Series Capacitor Compensated Transmission Networks.* PhD thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, 1980. 56, 57, 58, 60, 61, 71, 73, 155

[28] DJN Limebeer. Discrete-Time Optimal Control of a Synchronous Turbo-generator Under Transient Conditions. Master's thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, 1977. 56, 154

[29] The MathWorks. *MATLAB, The Language of Technical Computing.* The MathWorks. Inc., 24 Prime Park Way, Natick, MA, 5.3 edition, September 1998. 31, 72

[30] The MathWorks. *Control System Toolbox User Guide.* The MathWorks. Inc., 24 Prime Park Way, Natick, MA, 4.2 edition, January 1999. 78, 85, 163

[31] The MathWorks. *SIMULINK, Dynamic System Simulation for MATLAB.* The MathWorks. Inc., 24 Prime Park Way, Natick, MA, 3.0 edition, January 1999. 28

[32] Dallas Semiconductor MAXIM. Filter Design Using Integrator Blocks. Application Note, January 2001. http://www.maxim-ic.com. 109

[33] EJ Odendal. Micro-Machine Turbine Simulation. Master's thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, 1972. 30, 57, 59

[34] PS Rao and I Sen. Robust Tuning of Power System Stabilisers Using QFT. *IEEE Transactions on Control Systems Technology*, 7(4):478–485, July 1999. 1, 15, 16

[35] BS Rigby. Predicting and Measuring the Torsional Behaviour of Parallel Turbogenerators . Master's thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, 1992. 61

[36] MG Say. *Alternating Current Machines.* Longman Scientific and Technical, 1986. 23

[37] A Secker. *Modern Control of Synchronous Machines with Special Attention to Large Disturbances.* PhD thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, April 1981. 57

[38] AK Sedigh and G Alizadeh. Design of Power System Stabilisers Using Quantitative Feedback Theory. *IEEE Conference Publication*, (389):416–421, 1994. 9

[39] NM Shepstone. An Investigation into the Feasibility of using Artificial Neural Networks to Control Turbogenerators. Master's thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, January 1997.

[40] R Smith John. *Response Analysis of A.C. Machines.* John Wiley and Sons, 1990.

[41] T Smith Richard. *Analysis of Electrical Machines.* Pergamon Press, 1982. 20

[42] EAGLE Technology. PC26, PC30, and PC39 High Performance Analog I/O Boards. User Manual, January 1990. http://www.eagledaq.com. 63

[43] GK Venayagamoorthy. Implementation of a Continually Online Trained Artificial Neural Network Controller for a Turbogenerator. Master's thesis, Department of Electrical and Electronic Engineering, University of Natal, Durban, South Africa, 1998. 57

[44] Hoa Vu and J.C Agee. Comparison of Power System Stabilisers for Damping Local Mode Oscillations. *IEEE Transactions on Energy Conversion*, Vol.8(3):533–538, September 1993. 5, 8, 16

Appendix A

Machine Theory

A.1 Per Unit System and Sign Convention

The reciprocal base per unit system is used to convert all mechanical and electrical quantities to per unit values. Generally, the nominal rated values of voltage and current for the particular machine is chosen as the primary base values. However, the primary base values of voltage and current can be chosen arbitrarily. The per unit value of a primary quantity is expressed as a ratio of the actual value to the base value. For example, if the base value of voltage in a particular machine is 220V ac, and the actual voltage measured in the machine is 110V ac, then the per unit value of voltage in the machine is

$$\begin{aligned} V_{pu} &= \frac{V_{actual}}{V_{base}} \\ &= \frac{110}{220} \\ &= 0.5 \text{ p.u} \end{aligned} \tag{A.1}$$

The base values for secondary quantities, eg resistance, are derived from the primary base units. For example, in a transformer, if the ratio of the secondary winding to the primary winding is N, then the base value of secondary voltage is $N\times$(base value of primary voltage). Similarly, the secondary current base value is $\frac{1}{N}\times$(base value of primary current).

The per unit system is advantageous when performing design calculations for machines as it allows vastly different machines (in terms of ratings) to be compared quite easily. The magnitude of corresponding quantities are equal even if the machines have different designs. The per unit system also has the benefit that when inductances are converted to per unit, the number of turns of the coil does not enter into the equations [1], [21]. This simplifies the machine theory enormously, as the self inductance of coils can be found by simply adding the mutual and leakage inductances. Also, in terms of

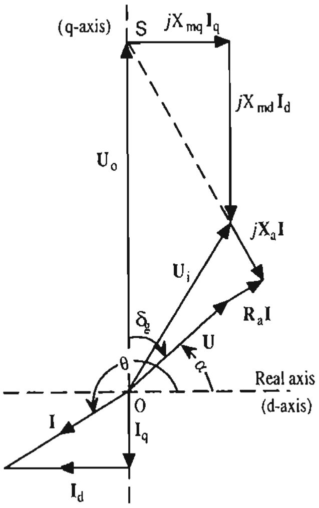


Figure A.1: Phasor diagram of a synchronous generator.

mechanical quantities, the torque is equal to the mechanical power. These relationships would not be true if the quantities were not in per unit.

The sign convention used for deriving the mathematical models and for calculation of initial values are as follows:

1. u represents the voltage applied to the circuit from an external source.

2. i represents the current flowing in the same direction as the applied voltage.

Therefore, in terms of the power flow, the instantaneous power $u \times i$ will flow into the circuit if both u and i are positive. This sign convention corresponds to motor operation. The Power System Blockset uses the generator convention, which means that power flows out of the circuit if both u and i are positive. Conversion from motor to generator convention is carried out by simply negating the signs of the equations.

A.2 Calculating Initial Conditions

The two-axis phasor diagram of a salient pole machine is shown in Figure A.1. The initial conditions are calculated from P_b, Q_b, and V_t. The equations to calculate the initial conditions are derived from the phasor diagram as shown in [1]:

$$
\begin{aligned}
\phi =& \arctan \frac{P_b}{Q_b} && \text{(A.2)}\\
I =& \sqrt{\frac{P_b^2 + Q_b^2}{V_b}} && \text{(A.3)}\\
\theta =& \arctan \left(\frac{I(X_q + X_t)\cos\phi - I(R_s + R_t)\sin\phi}{V_b + I(R_s + R_t)\cos\phi + I(X_q + X_t)\sin\phi} \right) && \text{(A.4)}\\
\gamma =& \frac{\pi}{2} - \theta + \phi && \text{(A.5)}\\
i_{do} =& I\cos\gamma && \text{(A.6)}\\
i_{qo} =& I\sin\gamma && \text{(A.7)}\\
v_{bd} =& V_b \sin\theta && \text{(A.8)}\\
v_{bq} =& V_b \cos\theta && \text{(A.9)}\\
v_{td} =& v_{bd} - i_{qo}X_t + i_{do}R_t && \text{(A.10)}\\
v_{tq} =& v_{bq} + i_{qo}R_t + i_{do}X_t && \text{(A.11)}\\
V_t =& \sqrt{v_{td}^2 + v_{tq}^2} && \text{(A.12)}\\
U_{od} =& v_{td} - i_{qo}X_q + i_{do}R_s && \text{(A.13)}\\
U_{oq} =& v_{tq} + i_{do}X_d + i_{qo}R_s && \text{(A.14)}\\
U_o =& \sqrt{U_{od}^2 + U_{oq}^2} && \text{(A.15)}\\
i_{fdo} =& \frac{U_o}{X_{md}} && \text{(A.16)}\\
P_m =& v_{td}i_{do} + v_{tq}i_{qo} + R_s I^2 && \text{(A.17)}\\
Q_m =& v_{td}i_{qo} - v_{tq}i_{do} && \text{(A.18)}\\
\lambda_{qo} =& -(L_l + L_{mq})i_{qo} && \text{(A.19)}\\
\lambda_{do} =& -(L_l + L_{md})i_{do} + L_{md}i_{fdo} && \text{(A.20)}\\
\lambda_{fdo} =& (L_{lfd} + L_{md})i_{fdo} - L_{md}i_{do} && \text{(A.21)}\\
\lambda_{kdo} =& L_{md}(-i_{do} + i_{fdo}) && \text{(A.22)}\\
\lambda_{kqo} =& -L_{mq}i_{qo} && \text{(A.23)}
\end{aligned}
$$

A.3 Machine Equations in Compact Form

The machine equations describing a synchronous generator can be written in terms of the currents rather than the flux linkages. This results in a compact representation if the equations are cast into state-space format. Equations A.24–A.31 gives the compact state-space representation of a synchronous generator model in terms of armature, field, and damper winding currents.

$$\boldsymbol{X} = \begin{bmatrix} -(L_{mq}+L_l) & 0 & 0 & 0 & L_{mq} \\ 0 & -(L_{md}+L_l) & L_{md} & L_{md} & 0 \\ 0 & -L_{md} & (L_{md}+L_{lfd}) & L_{md} & 0 \\ 0 & -L_{md} & L_{md} & (L_{md}+L_{lkd}) & 0 \\ -L_{mq} & 0 & 0 & 0 & (L_{mq}+L_{lkq}) \end{bmatrix} \tag{A.24}$$

$$\boldsymbol{W} = \begin{bmatrix} 0 & (L_{md}+L_l) & -L_{md} & -L_{md} & 0 \\ -(L_{mq}+L_l) & 0 & 0 & 0 & L_{mq} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.25}$$

$$\boldsymbol{R} = \begin{bmatrix} R_s & 0 & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 & 0 \\ 0 & 0 & -R_{fd} & 0 & 0 \\ 0 & 0 & 0 & -R_{kd} & 0 \\ 0 & 0 & 0 & 0 & -R_{kq} \end{bmatrix} \tag{A.26}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{i} = \boldsymbol{X}^{-1}(\boldsymbol{v} + (\boldsymbol{R} + \omega)\boldsymbol{W})\boldsymbol{i} \tag{A.27}$$

$$\lambda_q = -(L_{mq} + L_l)i_q + L_{mq}i_{kq} \tag{A.28}$$

$$\lambda_d = -(L_{md} + L_l)i_d + L_{md}(i_{fd} + i_{kd}) \tag{A.29}$$

$$T_e = \lambda_d i_q - \lambda_q i_d \tag{A.30}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\omega = \frac{1}{2H}(T_m - T_e) \tag{A.31}$$

Appendix B

Laboratory Equipment

B.1 Base Values

The base values of the laboratory electrical machinery are shown in Table B.1. Both the electrical and mechanical quantities are listed.

Base Values of Laboratory Generator		
Parameter	Value	Unit
Time	1	seconds
Speed	1	$\mathrm{rad\,s^{-1}}$
Armature Voltage	127	volts ac
Field Voltage	1231	volts dc
Armature Impedance	16.147	ohms
Field Power	1.50	kVA
Shaft Stiffness	38.2	$\frac{\mathrm{N \cdot m}}{\mathrm{rad\,s^{-1}}}$
No. of Poles	4	n/a
Angle	1	rad^e
Armature Power	1	kVA
Armature Current	7.87	amps ac
Field Current	1.218	amps dc
Field Impedance	1010.6	ohms
Torque	19.1	$\mathrm{N \cdot m}$
Inertia	38.2	$\mathrm{kg \cdot m^2}$

Table B.1: List of electrical and mechanical base values from [28].

B.2 Alternator Parameters

Table B.2 shows the alternator parameters as measured by various researchers. A "*" indicates that no measurement was available for that parameter.

Comparison of Generator Parameters			
Parameter	Limbeer[27]	Beck[4]	Fairbairn[13]
X_l	0.11	0.09	0.11
X_{ad}	1.98	2.49	1.98
X_{aq}	1.87	1.33	1.76
X_d	2.09	2.51	2.09
X_q	1.98	1.39	1.87
R_s	0.006	0.005	0.006
X_f	0.1	0.2124	0.0615
R_{fd}	0.00099	*	0.0011
X_{kd}	0.125	*	*
R_{kd}	0.0212	*	*
X_{kq}	0.257	*	*
R_{kq}	0.027	*	*

Table B.2: List showing the parameters as measured by various researchers.

Appendix C

Measured PSS Frequency Data

Measured Frequency Response		
Frequency(Hz)	Magnitude(dB)	Phase(deg)
0.1	0.0	54.0
0.2	0.0	57.0
0.3	6.0	71.0
0.4	8.0	63.0
0.5	9.5	65.0
0.6	14.0	65.0
0.7	14.0	65.0
0.8	15.0	58.0
0.9	15.0	63.0
1.0	18.0	62.0
2.0	20.0	43.0
3.0	23.5	36.0
4.0	24.6	23.0
5.0	24.7	0.0

Table C.1: The measured frequency response of the designed PSS.

Measured Frequency Response		
Frequency(Hz)	Magnitude(dB)	Phase(deg)
6.0	25.0	−64.0
7.0	24.7	−13.0
8.0	24.0	−19.5
9.0	23.6	−22.0
10.0	23.0	−28.0
20.0	19.0	−56.0
30.0	15.5	−69.0
40.0	13.0	−79.0
50.0	11.0	−75.0
60.0	10.0	−80.0
70.0	8.0	−83.0
80.0	6.0	−75.0
90.0	5.0	−84.0
100.0	3.5	−93.6

Table C.2: The measured frequency response of the designed PSS.

Appendix D

IEEE Excitation and Governor Systems

D.1 Excitation system

There are many types of excitation systems. Reference [21] gives the details of the various excitation systems in use today. In this study, only the static excitation system will be modelled, as it closely represents the exciter used in the laboratory experiments. The compound-source rectifier system exciter, or IEEE ST2A exciter as it is formally known, will be discussed. The power to this excitation system comes about by utilising the current, as well as the voltage of the generator output. It makes use of a saturable-current transformer and a power potential transformer. When the generator is not supplying a load, the power potential transformer supplies the entire excitation power supply. When the generator is loaded, some of the exciter power comes from the saturable-current transformer. This is useful under fault conditions, when the terminal voltage is close to zero. In this case, nearly all the excitation power is supplied by the current transformer. The physical representation of the exciter is shown in Figure D.1. When modelling the excitation system for small signal analysis, it is useful to cast Figure D.1 as a control block diagram shown in Figure D.2. The rectifier regulation block is represented in the expanded form as shown in Figure D.3. The equation describing the regulator as a function of excitation voltage is

$$\Delta E_{fd} = (\frac{K_c i_{fd}}{V_e})\Delta V_e \tag{D.1}$$

where K_c is specific for a particular machine. The amplifier block is modelled as a first order system with windup limits. As is the case with windups limits, the output of the system stays at the limit until the input to the system changes sign [21]. The block diagram is shown in Figure D.4. The final excitation system is shown in Figure D.5. This is a modified diagram com-

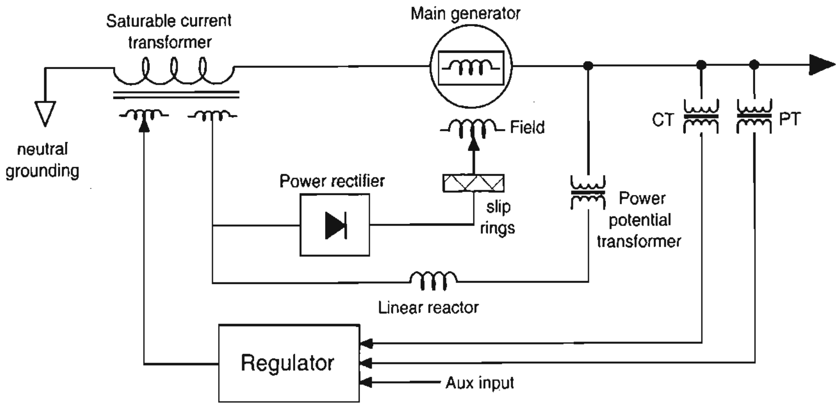


Figure D.1: Physical representation of a IEEE compound-source rectifier system exciter.

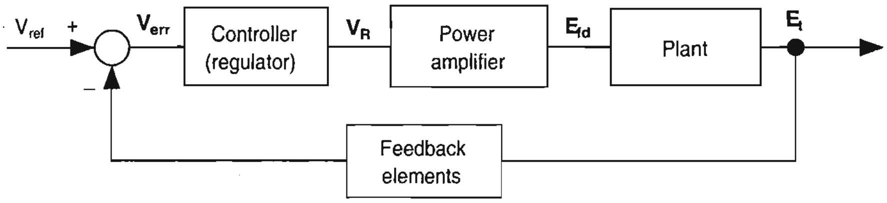


Figure D.2: Block diagram of the exciter suitable for small signal analysis.

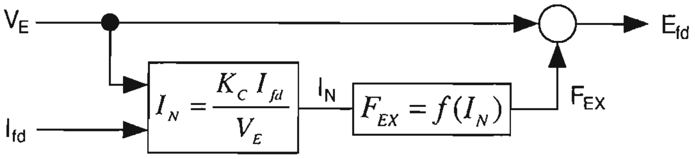


Figure D.3: Expanded view of the rectifier regulation block.

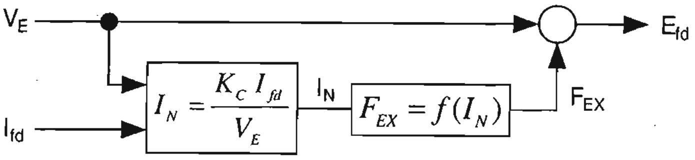


Figure D.4: Small signal model of the amplifier block.

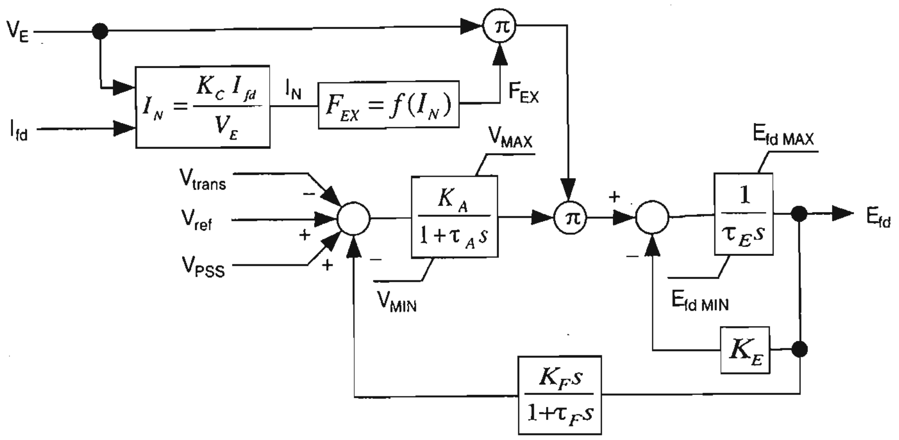


Figure D.5: Complete excitation model of the IEEE ST2A.

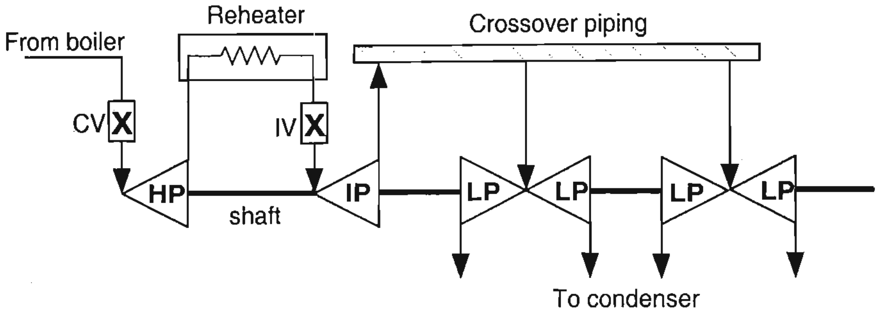


Figure D.6: Physical representation of a single reheat,tandem-compound steam turbine.

pared to the suggested ST2A model, but retains the essential components for accurate modelling.

D.2 Steam turbine and governor

A single reheat, tandem-compound turbine will be considered [21]. The governor is a mechanical-hydraulic controller. The steam turbine consists of a high-pressure turbine (HP), a intermediate pressure turbine (IP) and a low pressure turbine (LP). Figure D.6 shows the steam turbine components that will be considered for modelling. Steam from the boiler enters the HP section through the control valve (CV). The housing for the control valve is called the steam chest. The steam exiting the HP turbine is passed through a reheater. The reheated steam goes into the IP turbine via the reheat intercept valve (IV). The exhaust steam from the IP turbine section is fed to the inlet piping of the LP turbines via the crossover piping section. Figure D.7 shows the block diagram representation of the various turbine components. The response of steam flow to a change in control valve position has a time constant τ_{CH}. The reheater holds a substantial amount of steam, and it has a time constant τ_{RH} associated with it. The crossover piping has a time constant τ_{CO} that is accounted for in the model. If it is assumed that τ_{CO} is negligible compared to τ_{RH}, and the control valve characteristic is linear, then the simplified transfer function relating control valve position to turbine torque may be written as

$$\frac{\Delta T_m}{\Delta V_{CV}} = \frac{1 + F_{HP}\tau_{RH}s}{(1 + \tau_{CH}s)(1 + \tau_{RH}s)} \tag{D.2}$$

The functional block diagram of a mechanical hydraulic governor system for controlling a steam turbine is shown in Figure D.8. The governor consists of a speed governor, speed relays and hydraulic servomotors. The speed governor is a mechanical transducer that converts shaft speed to position. The speed governor output is amplified by a speed relay. The speed relay then drives the hydraulic servomotor that controls the valves on the steam turbine. A model representing the governor system for small deviations about an operating point is shown in Figure D.9. The speed governor is modelled by a gain constant K_G, the speed relay by a simple first order transfer function with time constant τ_{SR}, and the hydraulic servomotor by a non-windup integrator with time constant τ_{SM}.

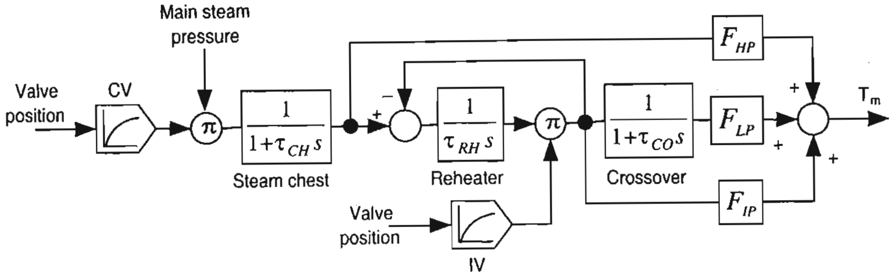


Figure D.7: Small signal model of the steam turbine suitable for analysis.

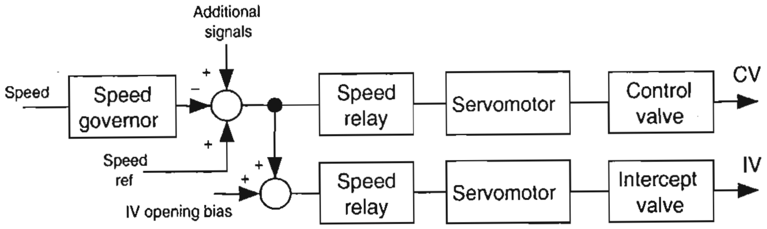


Figure D.8: The mechanical-hydraulic governor system in block diagram form.

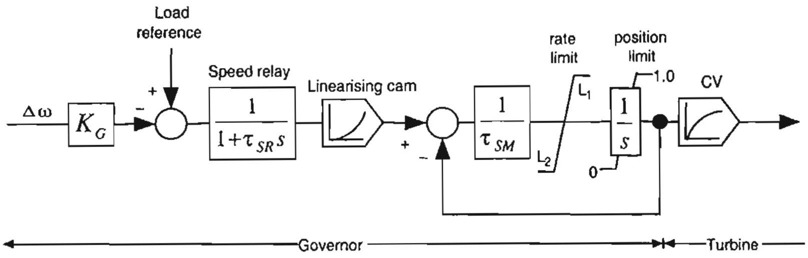


Figure D.9: Model of the mechanical-hydraulic governor suitable for small signal analysis.

Appendix E

Matlab's LTI Toolbox

This is a brief overview of Matlab's LTI Toolbox. For a thorough treatment on LTI models, refer to [30].

E.1 Creating LTI Models

LTI models may be created by using the `tf`, `zpk`, `ss`, and `frd` commands.

E.1.1 Transfer function

A transfer function model is created by 2 methods.

1. Use the `tf` command.

2. Express the transfer function in terms of the Laplace variable s.

For example, using the `tf` method, `L=tf([1,0],[1,2,10])` gives

$$L = \frac{s}{s^2 + 2s + 10} \tag{E.1}$$

To use the Laplace method, define `s=tf('s')`. Then Equation E.1 can be entered directly as `L=s/(s^2+2*s+10)`. MIMO systems may be formed by creating elementary SISO systems, then creating the MIMO LTI matrix. For example

$$H(s) = \begin{bmatrix} \frac{s-1}{s+1} \\ \frac{s+2}{s^2+4s+5} \end{bmatrix} \tag{E.2}$$

Define `h11=tf([1 -1],[1 1])` and `h21=tf([1 2],[1 4 5])`. Then $H(s)$ may be formed from `H=[h11;h21]`. To specify pure gain, use `tf` with a single argument e.g. `G=tf(4)` produces a `G` model with a gain of 4.

E.1.2 Zero-Pole-Gain models

ZPK models may be specified either by using the zpk command or by the Laplace variable s. For example the function

$$H(s) = \frac{-2s}{(s-2)(s^2-2s+2)} \tag{E.3}$$

can be entered in zpk format as

```
>>H=zpk([0],[1-i 1+i 2],[-2])
```

To use the Laplace method, define s=zpk('s') and enter the function symbolically. MIMO systems may be created by specifying H=[Z,P,K] where Z, P and K are P × M cell arrays of zeros, poles and gains respectively. For example,

$$H(s) = \begin{bmatrix} \frac{-1}{s} & \frac{3(s+5)}{(s+10^2)} \\ \frac{2(s^2-2s+2)}{(s-1)(s-2)(s-3)} & 0 \end{bmatrix} \tag{E.4}$$

is obtained by writing

```
>>Z={[],-5;[1-i 1+i],[]}
>>P={0,[-1 -1];[1 2 3],[]}
>>K=[-1,3;2,0]
>>H=zpk(Z,P,K)
```

E.1.3 State-space models

Continuous time models are of the form $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x} = \boldsymbol{Ax} + \boldsymbol{Bu}$ and $\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du}$ where $\boldsymbol{x}$ is the state vector, $\boldsymbol{u}$ is the input vector, and $\boldsymbol{y}$ is the output vector.

The state-space model is created by using >>sys=ss(A,B,C,D) where A is a $\boldsymbol{x} \times \boldsymbol{x}$ state matrix, B is a $\boldsymbol{x} \times \boldsymbol{u}$ input matrix, C is a $\boldsymbol{y} \times \boldsymbol{x}$ output matrix and D is a $\boldsymbol{y} \times \boldsymbol{u}$ throughput matrix.

E.1.4 Frequency response data models

A SISO frequency response data model is created by typing

```
>>sys=frd(response,frequency,units)
```

where response is a vector of complex frequency response values of length N, frequency is a real vector of length N, and units is 'rad/s' or 'Hz'.

A MIMO model is specified by a y × u × N multidimensional array for which response(y,u,k) is the complex frequency values from input u to output y at frequency frequency(k).

E.2 LTI Properties

The different LTI models share a common property structure called the generic property. Properties allow LTI models to be customised i.e names

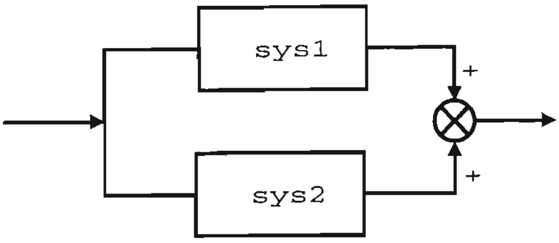


Figure E.1: The result of performing addition on two LTI models

can be assigned to inputs and outputs, notes can be stored in the model itself and delays can be programmed into the model.

For example, suppose there is a model `sys` defined, then the input names can be set as
`sys.inputname=['temp','pressure']`
There are also model specific properties, for example the `tf` structure uses the `num` and `den` properties.

Properties may also be set using the `set` command e.g
`set(sys,'inputname','temp',outputname,'pressure')`
A property value may be accessed by using the get function e.g
`PropVal=get(sys,'inputname')`

E.3 Operations on LTI Models

E.3.1 Subsystems

Subsystems may be extracted from models, e.g. `y=Hu` then `y3=H(3,1)` gives the relationship between the first input and the third input. The existing subsystem may also be replaced e.g `H(3,1)=NewSubSystem`. This redefines the relationship between the third output and first input. Indexing methods can also be applied. For example, to extract the subsystem relating the first 2 inputs to the 3rd output, use `subsys=sys(3,1:2)`.

E.3.2 Arithmetic operations

Almost all arithmetic operations can be applied to LTI models. Some of the operators are overloaded e.g the addition of 2 systems `sys=sys1+sys2` results in a parallel interconnection as shown in Figure E.1.

Subtraction results in the sign of the summing junction being changed. Multiplication, for example `sys=sys1*sys2`, results in the series connection of models as shown in Figure E.2.

Larger LTI models may be built up by using interconnection functions e.g

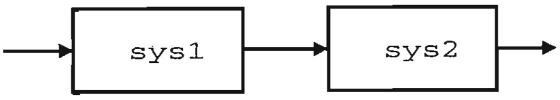


Figure E.2: The result of performing multiplication on two LTI models

`append` appends models in a block diagonal configuration, `connect` forms a state-space model from a block diagonal LTI object for an arbitrary interconnection matrix. `feedback` forms the feedback connection of 2 models.

E.4 Time and Frequency Response

System characteristics such as rise time, settling time, overshoot and steady state error can be determined from the time response. The `impulse` command can be used to generate an impulse response, `initial` can be used to generate an initial condition response, and `step` gives the step response of the system e.g `step(sys)`, `impulse(sys)`.

Frequency response utilities include bode plots, Nichols Charts, Nyquist diagrams, and Singular Valued plots. For example, `bode(sys,w)` where `w=logspace(-1,2,100)` is vector of logarithmic spaced frequencies.

Various responses for multiple models may be plotted e.g `bode(sys1,sys2)`. All models must have the same number of inputs and outputs. If the models are stored in an LTI array, then the responses of the array models may be plotted by typing `bode(LTIArrayName)`.

Appendix F

Program Listings

A list of the programs that are used in this thesis is given, along with a description of the program.

Matlab Program Listings	
Program	Description
mach_init	Constants file for laboratory equipment
qft_design	QFT plant iteration file
set_oppt	Sets the operating point for the linear model
lab_sys	Forms the state space matrices of a SM with transmission line connected to an infinite bus. Also includes the AVR and GOVERNOR.
setp	Calculates the initial conditions for the nonlinear PSB model connected to an infinite bus.
harley_init	Initialisation file for harley_gen
harley_gen	Simple nonlinear simulation of a synchronous machine.
harley_sfun	S-Function file for harley_gen

Table F.1: List of m-file program listings.

Matlab Program Listings	
Program	Description
igen_init	Initialisation file for igen_sfun.
igen_sfun	S-Function file for a generator using currents instead of fluxes.
krause_init	Initialisation file for krause_sfun.
krause_sfun	S-Function file for the machine equations given by [20].
ics	Single Machine Infinite Bus System Initial Conditions Solver.
lingen	Linearisation of the PSB equations.
shaftdyn_sfun	S-Function file for shaft dynamics.
fft	m-file for calculating FFT from data.
phigen	The flux linkage model.

Table F.2: List of m-file program listings (continued).

F.1 Machine Initialisation file

```
%****************************************
%* PROGRAM "mach_init"                  *
%* Constants file for lab equipment.    *
%*                                      *
%* Written by   Paramasivan Chetty      *
%*              Natal University        *
%*              09 September 2000       *
%*                                      *
%****************************************
%-------------------------------------
% GLOBALS
global Ll Lmd Lmq Ld Lq Lfd Lkd Lkq Lad Laq
global Llfd Llkd Llkq;
global Rs Rfd Rkd Rkq;
global wb H Kd;

%-------------------------------------
%NOMINAL VALUES
wb  = 2*pi*50;          %electrical freq in rad/sec

%-------------------------------------
%ELECTRICAL VARIABLES
Xl      = 0.11;         %p.u
Xmd     = 1.98;         %p.u
Xmq     = 1.87;         %p.u
Xd      = Xmd+Xl;       %p.u
Xq      = Xmq+Xl;       %p.u
Rs      = 0.006;        %p.u
Xf      = 0.1;          %p.u
Rfd     = 0.00099;      %p.u
Xkd     = 0.125;        %p.u
Rkd     = 0.0212;       %p.u
Xkq     = 0.257;        %p.u
Rkq     = 0.027;        %p.u

% In p.u, inductances are equal to reactances. (cf.Kundur)
Ll      = Xl;
Lmd     = Xmd;
Lmq     = Xmq;
Llfd    = Xf;
Llkd    = Xkd;
Llkq    = Xkq;
```

```
% Misc. inductances
Ld      = Ll+Lmd;
Lq      = Ll+Lmq;
Lfd     = Llfd+Lmd;
Lad     = 1/(1/Lmd + 1/Ll + 1/Llfd + 1/Llkd);
Laq     = 1/(1/Lmq + 1/Ll + 1/Llkq);

%--------------------------------------
%MECHANICAL VARIABLES
H       = 5.6814;
Kd      = 0;

%--------------------------------------
%AVR TIME CONSTANTS
Tv1     = 0.616;
Tv2     = 2.178;
Tv3     = 0.199;
Tv4     = 0.039;
Tv5     = 0.0235;
Te      = 0.49;
Rf      = 47;               %Rf = 47 Kohms max.
Ka      = 4.38*Rf;          % cf.Limbeer (MSc)

%--------------------------------------
%GOVERNOR TIME CONSTANTS
Tg1     = 0.3;
Tg2     = 0.03;
Tg3     = 0.15;
Tg4     = 0.62;
Tg5     = 2.56;
F       = 0.337;
Rg      = 100;              %Rg = 100 Kohms max.
Kg      = 9.63e-4*Rg;       % cf.Limbeer (MSc)
```

F.2 QFT Design file

```
%**************************************
% PROGRAM "qft_design"                *
% Plant iteration file                *
%                                     *
% PLANT:                              *
% Ef--->|t11 t12|--->Vt               *
%       |   T   |                     *
% Tm--->|t21 t22|--->w                *
%                                     *
%**************************************
%--------------------------------------
% DATA
Vb=1.0;
%case 1 2 3 4 5 6 7 8 9
P=[0.9 0.05 0.5 0.05 0.1 0.3 0.7 0.9 0.9];
Q=[0.3 0.0 -0.2 0.0 0.0 0.0 0.4 0.5 -0.4];
XT=[0.542 0.542 0.542 0.126 0.126 0.542 0.542 0.249...
    0.0675];
RT=[0.0128 0.0128 0.0128 0.00713 0.00713 0.0128 0.0128...
    0.00961 0.00471];
%freq range
W=[1 2 3 4 5 5.5 6 6.5 7 7.5 8 8.5 9 9.5 10 15 20];

%--------------------------------------
% WASHOUT
NUMCO=[10 0];           % sTw/(sTw + 1)
DENCO=[10 1];

%--------------------------------------
% ITERATION
EndVal=length(P);
for Count = 1:EndVal,
   Pb   = P(Count);
   Qb   = Q(Count);
   Xt   = XT(Count);
   Rt   = RT(Count);
   set_oppt; %calculate initial conditions for cases
   lab_sys;  %generate plant state space model

   TArray(:,:,Count)=SYS;
   GArray(:,:,Count)=SYS;
   T21  = SYS(2,1);      %delta Ef to delta w
   [num,den]=tfdata(T21,'v');
   N21(Count,:)=num;     %numerator coefficients
   D21(Count,:)=den;     %denominator coefficients

   T22  = SYS(2,2);      %delta Tm to delta w
   [num,den]=tfdata(T22,'v');
   N22(Count,:)=num;
   D22(Count,:)=den;
end;

%---------------------------------------
% NOMINAL PLANT
NUMPO   = N21(1,:);
DENPO   = D21(1,:);

%---------------------------------------
% BODE PLOTS
figure;%bode of delta Ef to delta w
bode(TArray(2,1,:),{0.1 100});
figure;%bode of delta Tm to delta w
bode(TArray(2,2,:),{0.1 100});

%---------------------------------------
% PLANT TEMPLATES
phs=(-360:3:0);
f22=freqcp(N22,D22,W);
f220=freqcp(N22,D22,1);%Calc. T11 @ w=1 rad/sec.
f21=freqcp(N21,D21,W);
plottmpl(W,W,f21,1);
title('Plant Templates');

%---------------------------------------
% BOUNDS
disp('Generating bounds');
% |1/(1+L)|<=3dB
% SISOBNDS(2,W,W,WS,P,[],NOM,[],[],PHS)
WS=10^(3/20);
NOM=1;
bnd0=sisobnds(2,W,[],WS,f21,[],NOM,[],1,phs);
plotbnds(bnd0,[],phs);
title('Stability Bounds');
sbndi=sectbnds(bnd0);
```

```
% |T11/(1+L)|<=(6dB + |T11|@w=0)
% GENBNDS(10,W,WBD,WS,A,B,C,D,Pnom,PHS)
Count=0;
A=f22;
B=0;
C=1;
D=f21;
Pnom=f21(1,:);
bnds=bnd0;
pbnds=[];
for Count=1:EndVal,
WS= 10^(6/20) * abs(f220(Count,1));
bnd=genbnds(10,W,W,WS,A,B,C,D,Pnom,phs);
bnds=[bnds,bnd];          %group bounds
pbnds=[pbnds,bnd];        %performance bounds
end;

%-------------------------------------------
% PLOT BOUNDS
% PLOTBNDS(BNDS,PTYPE,PHS)
plotbnds(bnds,[],phs);
title('All bounds');
pbnd    = sectbnds(pbnds);
plotbnds(pbnd,[],phs);
title('Performance Bounds');

%-------------------------------------------
% BOUNDARY INTERSECTION
bndi     = sectbnds(bnds);

%-------------------------------------------
% LOOP SHAPING
% LPSHAPE(W,BDB,NUMPO,DENPO,DELAY,NUMCO,DENCO,PHS)
lpshape(W,bndi,-NUMPO,DENPO,0,NUMCO,DENCO,phs);
```

F.3 Operating Point file

```
%******************************************
%* PROGRAM "set_oppt"                     *
%* Sets the operating point for the       *
%* linear model.                          *
%*                                        *
%* Written by   Paramasivan Chetty        *
%*              Natal University          *
%*              19 October 2000           *
%*                                        *
%******************************************
%------------------------------------------
% CALCULATIONS
phi     = angle(Pb+j*Qb);
I       = sqrt(Pb^2+Qb^2)/Vb;
tho     = atan((I*(Xq+Xt)*cos(phi)-I*(Rs+Rt)*sin(phi))...
            /(Vb+I*(Rs+Rt)*cos(phi)+I*(Xq+Xt)*sin(phi)));
gam     = pi/2 - (tho + phi);
ido     = I*cos(gam);
iqo     = I*sin(gam);
vbd     = Vb*sin(tho);
vbq     = Vb*cos(tho);
vtd     = vbd - iqo*Xt + ido*Rt;
vtq     = vbq + iqo*Rt + ido*Xt;
Vt      = sqrt(vtd^2 + vtq^2);
Uod     = vtd - iqo*Xq + ido*Rs;
Uoq     = vtq + ido*Xd + iqo*Rs;
Uo      = sqrt(Uod^2 + Uoq^2);
ifdo    = Uo/Xmd;
Pm      = vtd*ido + vtq*iqo + (Rs)*I^2;
Qm      = vtd*iqo - vtq*ido;

dwo     = 0;
phiqo   = -(Ll+Lmq)*iqo;
phido   = -(Ll+Lmd)*ido + Lmd*ifdo;
phifdo  = (Llfd+Lmd)*ifdo - Lmd*ido;
phikdo  = Lmd*(-ido+ifdo);
phikqo  = -Lmq*iqo;
```

F.4 Laboratory System file

```
%*****************************************
%* PROGRAM "lab_sys"                     *
%* Forms the state space matrices of a   *
%* SM with transmission line connected   *
%* to an infinite bus.Also includes the  *
%* AVR and GOVERNOR.                     *
%*                                       *
%* Written by P Chetty                   *
%* University of Natal                   *
%* 19 October 2000                       *
%*                                       *
%*****************************************
%-----------------------------------------
%DEFINE VARIABLES
wro     = 0;

%-----------------------------------------
%GENERATOR

%delta pphid
FD1 = wb*(Rs+Rt)/Ll*(Lad/Ll - 1);              %delta phid
FD2 = wb*(Xt/Ll-Xt*Laq/Ll^2 + 1 + wro);        %delta phiq
FD3 = wb*(Rs+Rt)*Lad/Llfd/Ll;                  %delta phifd
FD4 = wb*(Rs+Rt)*Lad/Llkd/Ll;                  %delta phikd
FD5 = -wb*Xt*Laq/Ll/Llkq;                      %delta phikq
FD6 = wb*(Xt*iqo+phiqo);                       %delta wr
FD7 = wb*vbq;                                  %delta theta

%delta pphiq
FQ1 = wb*(Xt*Lad/Ll^2-Xt/Ll-1-wro);            %delta phid
FQ2 = wb*(Rs+Rt)/Ll*(Laq/Ll - 1);              %delta phiq
FQ3 = wb*Xt*Lad/Ll/Llfd;                       %delta phifd
FQ4 = wb*Xt*Lad/Ll/Llkd;                       %delta phikd
FQ5 = wb*(Rs+Rt)*Laq/Llkq/Ll;                  %delta phikq
FQ6 = wb*(Xt*ido-phido);                       %delta wr
FQ7 = -wb*vbd;                                 %delta theta

%delta pphifd
U1  = wb*Rfd/Lmd;                              %delta vfd
FF1 = wb*Rfd*Lad/Llfd/Ll;                      %delta phid
FF2 = 0;                                       %delta phiq
FF3 = wb*Rfd/Llfd*(Lad/Llfd - 1);              %delta phifd
```

```
FF4 = wb*Rfd*Lad/Llfd/Llkd;                 %delta phikd
FF5 = 0;                                    %delta phikq
FF6 = 0;                                    %delta wr
FF7 = 0;                                    %delta theta

%delta pphikd
FKD1    = wb*Rkd*Lad/Llkd/Ll;               %delta phid
FKD2    = 0;                                %delta phiq
FKD3    = wb*Rkd*Lad/Llkd/Llfd;             %delta phifd
FKD4    = wb*Rkd/Llkd*(Lad/Llkd - 1);       %delta phikd
FKD5    = 0;                                %delta phikq
FKD6    = 0;                                %delta wr
FKD7    = 0;                                %delta theta

%delta pphikq
FKQ1    = 0;                                %delta phid
FKQ2    = wb*Rkq*Laq/Llkq/Ll;               %delta phiq
FKQ3    = 0;                                %delta phifd
FKQ4    = 0;                                %delta phikd
FKQ5    = wb*Rkq/Llkq*(Laq/Llkq - 1);       %delta pphikq
FKQ6    = 0;                                %delta wr
FKQ7    = 0;                                %delta theta

%delta pwr
U2      = 1/2/H;                            %delta pm
W1      = (-phiqo/Ll*(1-Lad/Ll)-iqo)/2/H;   %delta phid
W2      = (phido/Ll*(1-Laq/Ll)+ido)/2/H;    %delta phiq
W3      = (phiqo*Lad/Llfd/Ll)/2/H;          %delta phifd
W4      = (phiqo*Lad/Llkd/Ll)/2/H;          %delta phikd
W5      = -(phido*Laq/Llkq/Ll)/2/H;         %delta phikq
W6      = -Kd;                              %delta wr
W7      = 0;                                %delta theta

%delta theta
TH1 = 0;                                    %delta phid
TH2 = 0;                                    %delta phiq
TH3 = 0;                                    %delta phifd
TH4 = 0;                                    %delta phikd
TH5 = 0;                                    %delta phikq
TH6 = wb;                                   %delta wr
TH7 = 0;                                    %delta theta

%ABCD MATRICES
```

```
%form A
A=[FD1  FD2     FD3     FD4     FD5     FD6     FD7;
   FQ1  FQ2     FQ3     FQ4     FQ5     FQ6     FQ7;
   FF1  FF2     FF3     FF4     FF5     FF6     FF7;
   FKD1 FKD2    FKD3    FKD4    FKD5    FKD6    FKD7;
   FKQ1 FKQ2    FKQ3    FKQ4    FKQ5    FKQ6    FKQ7;
   W1   W2      W3      W4      W5      W6      W7;
   TH1  TH2     TH3     TH4     TH5     TH6     TH7];

%form B
% vfd pm
B=[0        0;
   0        0;
   U1       0;
   0        0;
   0        0;
   0        U2;
   0        0];

%form C
%v_qd output
VQ1 = -Xt/Ll*(1-Lad/Ll);                %delta phid
VQ2 = -Rt/Ll*(1-Laq/Ll);                %delta phiq
VQ3 = Xt*Lad/Ll/Llfd;                   %delta phifd
VQ4 = Xt*Lad/Ll/Llkd;                   %delta phikd
VQ5 = Rt*Laq/Ll/Llkq;                   %delta phikq
VQ6 = Xt*ido;                           %delta wr
VQ7 = -vbd;                             %delta theta

VD1 = -Rt/Ll*(1-Lad/Ll);                %delta phid
VD2 = Xt/Ll*(1-Laq/Ll);                 %delta phiq
VD3 = Rt*Lad/Ll/Llfd;                   %delta phifd
VD4 = Rt*Lad/Ll/Llkd;                   %delta phikd
VD5 = -Xt*Laq/Ll/Llkq;                  %delta phikq
VD6 = Xt*iqo;                           %delta wr
VD7 = vbq;                              %delta theta

%delta Vt
VT1 = vtd/Vt*VD1 + vtq/Vt*VQ1;          %delta phid
VT2 = vtd/Vt*VD2 + vtq/Vt*VQ2;          %delta phiq
VT3 = vtd/Vt*VD3 + vtq/Vt*VQ3;          %delta phifd
VT4 = vtd/Vt*VD4 + vtq/Vt*VQ4;          %delta phikd
VT5 = vtd/Vt*VD5 + vtq/Vt*VQ5;          %delta phikq
VT6 = vtd/Vt*VD6 + vtq/Vt*VQ6;          %delta wr
```

```
VT7 = vtd/Vt*VD7 + vtq/Vt*VQ7;       %delta theta


C=[VT1  VT2 VT3 VT4 VT5 VT6 VT7;     %delta vtq
   0    0   0   0   0   1   0];      %delta wr

%form D
D   = zeros(2,2);

%form GEN
inputs  ={'Rvt' 'Rtm'};
outputs ={'vt' 'wr'};
states ={'phid' 'phiq' 'phifd' 'phikd' 'phikq' 'wr' 'th'};
GEN     = ss(A,B,C,D,'statename',states,'inputname',...
          inputs,'outputname',outputs);

%-------------------------------------------
%AVR STATE SPACE
pnum     = conv([Tv1 1],[Tv3 1]);
pden     = conv([Tv2 1],[Tv4 1]);
PID = tf(pnum*Ka,pden);
exc = tf(1,[Te 1]);
sys = series(PID,exc);
AVR = ss(sys);
set(AVR,'inputname',{'eVt'},'outputname',{'vf'});
[AVR_A,AVR_B,AVR_C,AVR_D]=ssdata(AVR);

%-------------------------------------------
%TERMINAL VOLTAGE TRANSDUCER TVT
TVT = ss(tf(1,[Tv5 1]));
set(TVT,'inputname',{'Fvt'},'outputname',{'Mvt'});
[TVT_A,TVT_B,TVT_C,TVT_D]=ssdata(TVT);

%-------------------------------------------
%GOVERNOR STATE SPACE
servo    = tf(1,[Tg3 1]);
ent      = tf(1,[Tg4 1]);
rheat    = tf([F*Tg5 1],[Tg5 1]);
sys1     = series(servo,ent);
sys      = series(sys1,rheat);
GOV      = ss(sys);
set(GOV,'inputname',{'eTm'},'outputname',{'tm'});
[GOV_A,GOV_B,GOV_C,GOV_D]=ssdata(GOV);
```

```
%------------------------------------------
%SPEED TRANSDUCER ST
ST      = ss(tf([Tg1 1]*Kg,[Tg2 1]));
set(ST,'inputname',{'Fwr'},'outputname',{'Mtm'});
[ST_A,ST_B,ST_C,ST_D]=ssdata(ST);

%------------------------------------------
%FORM SYSTEM
AVRGOV = append(AVR,GOV);
AGG    = series(AVRGOV,GEN);
TVTST  = append(TVT,ST);
SYS    = feedback(AGG,TVTST);
set(SYS,'inputname',{'Vref' 'Pref'},...
    'outputname',{'Vt' 'w'});
[SYS_A,SYS_B,SYS_C,SYS_D]=ssdata(SYS);
```

F.5 Nonlinear Operating Point file

```
%*****************************************
%* PROGRAM "setp"                        *
%* Calculates the initial conditions     *
%* for the nonlinear PSB model connected *
%* to an infinite bus.Also includes the  *
%* AVR and GOVERNOR.                     *
%*                                       *
%* Written by P Chetty                   *
%* University of Natal                   *
%* 29 November 2001                      *
%*                                       *
%*****************************************
%----------------------------------------
% INITIALISATION
Xd  = 2.09;
Xq  = 1.98;
Xa  = 0.11;
Rs  = 0.006;         %Ra
Vb  = 1.00;
Pb  = 0.9;
Qb  = 0.3;
Xt  = (0.1295  + 0.542);
Rt  = (0.0071  + 0.0128);

%----------------------------------------
% CALCULATIONS

phi = angle(Pb+j*Qb);
I   = sqrt(Pb^2+Qb^2)/Vb;
tho = atan((I*(Xq+Xt)*cos(phi)-I*(Rs+Rt)*sin(phi))/...
        (Vb+I*(Rs+Rt)*cos(phi)+I*(Xq+Xt)*sin(phi)));
gam = pi/2 - (tho + phi);
ido = I*cos(gam);
iqo = I*sin(gam);
vbd = Vb*sin(tho);
vbq = Vb*cos(tho);
vtd = vbd - iqo*Xt + ido*Rt;
vtq = vbq + iqo*Rt + ido*Xt;
Vt  = sqrt(vtd^2 + vtq^2);
Uod = vtd - iqo*Xq + ido*Rs;
Uoq = vtq + ido*Xd + iqo*Rs;
Uo  = sqrt(Uod^2 + Uoq^2);
```

```
Pm  = vtd*ido + vtq*iqo + Rs*I^2;
Qm  = vtd*iqo - vtq*ido;
pwr_ang = phi*180/pi;
rot_ang = tho*180/pi;
```

F.6 Flux Model file

```
%********************************************
%* PROGRAM "phi_gen"                        *
%* M-FILE for phigen                        *
%*                                          *
%* Written by P Chetty                      *
%* University of Natal                      *
%* 11 September 2000                        *
%*                                          *
%********************************************
function[out_vector]    = phi_gen(in_vector);
%--------------------------------------------
% GLOBALS
global Ll Lad Laq Lmd Llfd Llkd Llkq Xt;
global Rs Rfd Rkd Rkq Rt;
global wb H Kd;

%--------------------------------------------
% BEGIN
phid     = in_vector(1);
phiq     = in_vector(2);
phifd    = in_vector(3);
phikd    = in_vector(4);
phikq    = in_vector(5);
wr       = in_vector(6);
theta    = in_vector(7);
vd       = in_vector(9);
vq       = in_vector(8);
pm       = in_vector(10);
vfd      = in_vector(11);

phimd    = Lad*(phid/Ll + phifd/Llfd + phikd/Llkd);
phimq    = Laq*(phiq/Ll + phikq/Llkq);
id       = -1/Ll*(phid - phimd);
iq       = -1/Ll*(phiq - phimq);

vtd      = vd - iq*Xt + id*Rt;
vtq      = vq + iq*Rt + id*Xt;

pphid    = wb*(vtd + (1+wr)*phiq + Rs/Ll*(phimd - phid));
pphiq    = wb*(vtq - (1+wr)*phid + Rs/Ll*(phimq - phiq));
pphifd   = wb*(vfd*Rfd/Lmd + Rfd/Llfd*(phimd - phifd));
pphikd   = wb*(Rkd/Llkd*(phimd - phikd));
```

```
pphikq   = wb*(Rkq/Llkq*(phimq - phikq));
Te       = phid*iq - phiq*id;
pwr      = (1/2/H)*(pm - Te - Kd*(wr));
ptheta   = wb*(wr);

%-------------------------------------------
% OUTPUT FORMATTING
out_vector  = [pphid pphiq pphifd pphikd pphikq pwr...
               ptheta iq id vtq vtd];
```

F.7 Nonlinear SM Initialisation file

```
%*****************************************
%* PROGRAM "harley_init"                 *
%* Initialisation file for harley_gen    *
%*                                       *
%* Written by   Paramasivan Chetty       *
%*              Natal University         *
%*              29 August 2000           *
%*                                       *
%*****************************************
%---------------------------------------------
%INITIALISATION
global D Q Ra Rf Rkd Rkq J w0 P Kd H;
global Vtd Vtq Vb fd0 fq0 ff0 fkd0 fkq0 del;
w0 = 1.0;

%---------------------------------------------
%MACHINE PARAMETERS
Xmd = 1.98;
Xmq = 1.87;
Xa  = 0.11;
Ra  = 0.006;
Xf  = 0.1;
Rf  = 0.00099;
Xkd = 0.125;
Rkd = 0.0212;
Xkq = 0.257;
Rkq = 0.027;
J   = 2.355e-3 + 7.608e-3 + 7.576e-3 + 7.779e-3 +...
      1.063e-2 + 2.192e-5;
XD  = [Xmd+Xa,  Xmd,     Xmd;     Xmd,     Xmd+Xf, Xmd;...
       Xmd,     Xmd,     Xmd+Xkd];
XQ  = [Xmq+Xa, Xmq; Xmq, Xmq+Xkq];
D   = inv(XD);
Q   = inv(XQ);
P   = 2;
Kd  = 0;
H   = 5.6;

%---------------------------------------------
%INITIAL CONDITIONS
% dialog box
```

```
prompt={'Bus voltage Vb (p.u)','Active power Pb (p.u)'...
,'Reactive power Qb (p.u)','Line reactance (p.u)'...
,'Line resistance (p.u)','Base impedance (ohms)'};
if exist('Vb','var');
   sVb=num2str(Vb);
else
   sVb='1.0';
end
if exist('Pb','var');
   sPb=num2str(Pb);
else
   sPb='0.0';
end
if exist('Qb','var');
   sQb=num2str(Qb);
else
   sQb='0.0';
end
if exist('Xt','var');
   sXt=num2str(Xt);
else
   sXt='0.0';
end
if exist('Rt','var');
   sRt=num2str(Rt);
else
   sRt='0.0';
end
if exist('Zb','var');
   sZb=num2str(Zb);
else
   sZb='16.14';
end
def={sVb,sPb,sQb,sXt,sRt,sZb};
dlgTitle='Set SM operating point';
lineNo=1;
answer=inputdlg(prompt,dlgTitle,lineNo,def);
if isempty(answer);
   return
end
Vb=str2double(answer{1});
if isnan(Vb);
   msgbox('Value for bus voltage is not a...
         valid number','Bus voltage','error');
   return
end
Pb=str2double(answer{2});
if isnan(Pb);
   msgbox('Value for active power is not a...
         valid number','Active power','error');
   return
end
Qb=str2double(answer{3});
if isnan(Qb);
   msgbox('Value for reactive power is not a...
         valid number','Reactive power','error');
   return
end
Xt=str2double(answer{4});
if isnan(Xt);
   msgbox('Value for line reactance is not a...
         valid number','Transmission line','error');
   return
end
Rt=str2double(answer{5});
if isnan(Rt);
   msgbox('Value for line resistance is not a...
         valid number','Transmission line','error');
   return
end
Zb=str2double(answer{6});
if isnan(Zb);
   msgbox('Value for base impedance is not a...
         valid number','Base impedance','error');
   return
end

%-----------------------------------------------
%CALCULATIONS
Xd       = 2.09;
Xq       = 1.98;
I        = (sqrt(Pb^2+Qb^2))/Vb;
phi      =angle(Pb+j*Qb);
del      = atan((I*(Xq+Xt)*cos(phi)-I*(Ra+Rt)*sin(phi))...
             /(Vb+I*(Ra+Rt)*cos(phi)+I*(Xq+Xt)*sin(phi)));
gam      = pi/2 - (del + phi);
Id       = I*cos(gam);
Iq       = I*sin(gam);
```

```
Vbd      = Vb*sin(del);
Vbq      = Vb*cos(del);
Vtd      = Vbd + Iq*Xt - Id*Rt;
Vtq      = Vbq - Iq*Rt - Id*Xt;
Vt       = sqrt(Vtd^2 + Vtq^2);
Uod      = Vtd + Iq*Xq - Id*Ra;
Uoq      = Vtq - Id*Xd - Iq*Ra;
Uo       = sqrt(Uod^2 + Uoq^2);
If       = (Uo*sqrt(2))/Xmd;
Vf       = If*Rf;
Ikd      = -Id + If;
Ikq      = Iq;
Pm       = (Vtd*Id + Vtq*Iq);
Qm       = Vtd*Iq - Vtq*Id;
fq0      = (Xmq+Xa)*Iq;
fkq0     = (Xmq)*Iq;
fd0      = -(Xmd+Xa)*Id+(Xmd)*If;
fkd0     = (Xmd)*(-Id+If);
ff0      = (Xf+Xmd)*If + (Xmd)*(-Id);

%--------------------------------------------
%DISPLAY RESULTS
clc;
header=sprintf('Steady state values with inputs...
Vb=%4.4f p.u, Pb=%4.4f p.u and Qb=%4.4f p.u',Vb,Pb,Qb);
wIf=sprintf('If =   %f  %f  amps(1.218)',If,If*1.218);
wVf=sprintf('Vf =   %f  %f  volts(1231)',Vf,Vf*1231);
wId=sprintf('Id0    =   %f  %f  amps(7.87)',Id,Id*7.87);
wIq=sprintf('Iq0    =   %f  %f  amps(7.87)',Iq,Iq*7.87);
wVd=sprintf('Uod    =   %f  %f  volts(220)',Uod,Uod*220);
wVq=sprintf('Uoq    =   %f  %f  volts(220)',Uoq,Uoq*220);
wphi=sprintf('phi   =   %f          degrees',phi*180/pi);
wdel=sprintf('del   =   %f          degrees',del*180/pi);
wgam=sprintf('gam   =   %f          degrees',gam*180/pi);
wIt=sprintf('I  =   %f  %f  volts(220)',I,I*7.87);
wVt=sprintf('Vt =   %f  %f  volts(220)',Vt,Vt*220);
wPm=sprintf('Pm =   %f  %f  watts(1000)',Pm,Pm*1e3);
wQm=sprintf('Qm =   %f  %f  vars(1000)',Qm,Qm*1e3);
wUo=sprintf('Uo =   %f  %f  volts(220)',Uo,Uo*220);

disp(header);
disp(' ');
disp('Parameter   p.u        actual     unit(base)');
disp(' ');
disp(wIf);
disp(wVf);
disp(wId);
disp(wIq);
disp(wVd);
disp(wVq);
disp(wphi);
disp(wdel);
disp(wgam);
disp(wVt);
disp(wIt);
disp(wPm);
disp(wQm);
disp(wUo);
```

F.8 Nonlinear SM Execution file

```
%*******************************************************
%* PROGRAM "harley_gen"                                *
%* Simple nonlinear simulation of a synchronous machine. *
%*                                                     *
%* This m-file simulates the nonlinear model of a SMIB. *
%* The model is assumed to be connected to an infinite  *
%* bus without a transmission line.                    *
%* The general synchronous machine equations derived   *
%* from Adkins and Harley are to be solved.            *
%* Saturation effects can be modelled by modifying the *
%* inductance but this has limited advantages.         *
%*                                                     *
%* The following notation is used:                     *
%* (p is the derivative operator.)                     *
%* fd,ff,fkd,Rf,Rkd = d-axis fluxes and resistances.   *
%* fq,fkq,Rkq       = q-axis fluxes and resistance     *
%* ud,uq,uf,ukd,ukq = axes voltages                    *
%* D,Q,Ra           = stator reactances and resistance *
%* Tm,w0,J,Kd       = applied torque, sync speed,      *
%*                    inertia and damping constant.    *
%* del              = rotor angle                      *
%* All values are in per unit.                         *
%*                                                     *
%* Written by   Paramasivan Chetty                     *
%*              Natal University                       *
%*              29 August 2000                         *
%*                                                     *
%*******************************************************
function[pOut]=harley_gen(var_vector);
%---------------------------------------------
%INITIALISATION
global D Q Ra Rf Rkd Rkq J w0 P Kd H;
fd      = var_vector(1);
fq      = var_vector(2);
ff      = var_vector(3);
fkd     = var_vector(4);
fkq     = var_vector(5);
ud      = var_vector(6);
um      = var_vector(7);
uf      = var_vector(8);
Tm      = var_vector(9);
pdel    = var_vector(11);
```

```
del      = var_vector(10);

%-----------------------------------------------
%CALCULATIONS
% electrical equations
ud  = um*sin(del);
uq  = um*cos(del);
pfd = ud-fq-Ra*(D(1,1)*fd+D(1,2)*ff+D(1,3)*fkd)+...
                                  fq*pdel/w0;
pfq = uq+fd-Ra*(Q(1,1)*fq+Q(1,2)*fkq)-fd*pdel/w0;
pff = uf-Rf*(D(2,1)*fd+D(2,2)*ff+D(2,3)*fkd);
pfkd= -Rkd*(D(3,1)*fd+D(3,2)*ff+D(3,3)*fkd);
pfkq= -Rkq*(Q(2,1)*fq+Q(2,2)*fkq);
Id  = D*[fd;ff;fkd];
Iq  = Q*[fq;fkq];
i   = [Id(1) Iq(1)];
v   = [ud uq];

% mechanical equation
ppdel    = (1/J)*((1/(4))*(fd*fq*(Q(1,1)-D(1,1))+Q(1,2)...
     *fd*fkq-D(1,2)*fq*ff-D(1,3)*fq*fkd) - Tm - Kd*pdel);

%-----------------------------------------------
%OUTPUTS
pOut     = [pfd,pfq,pff,pfkd,pfkq,ppdel,i,v];
```

F.9 S-Function of Nonlinear SM file

```
%*******************************************
%* PROGRAM "harley_sfun"                   *
%* S-Function file for harley_gen          *
%*                                         *
%* Written by   Paramasivan Chetty         *
%*              Natal University           *
%*              2 August 2000              *
%*                                         *
%*******************************************
function [sys,x0,str,ts] = harley_sfun(t,x,u,flag)
%-----------------------------------------------
% DEFINE VARIABLES
global D Q Ra Rf Rkd Rkq J w0 P Kd H;
global Vtd Vtq Vb fd0 fq0 ff0 fkd0 fkq0 del phi;

%-----------------------------------------------
% BEGIN
switch flag,

case 0,
 sys =[7 0 4 2 1 0 1];
 x0  =[0 0 0 0 0 0 0];
 str =[];
 ts  =[0 0];

case 1,
 fd=x(1);fq=x(2);ff=x(3);fkd=x(4);fkq=x(5);
 pang=x(6);ang=x(7);
 uf  =u(1);
 Tm  =u(2);
 % DQ transformation
 ud  =(2*Vb/3)*(cos(ang)+cos(ang-2*pi/3)+cos(ang-4*pi/3));
 uq  =(2*Vb/3)*(sin(ang)+sin(ang-2*pi/3)+sin(ang-4*pi/3));
 pfd =ud-w0*fq-Ra*(D(1,1)*fd+D(1,2)*ff+D(1,3)*...
                                 fkd)*w0+fq*pang;
 pfq = uq + w0*fd - Ra*(Q(1,1)*fq + Q(1,2)*...
                            fkq)*w0 - fd*pang;
 pff = uf - Rf*(D(2,1)*fd + D(2,2)*ff + D(2,3)*fkd)*w0;
 pfkd= -Rkd*(D(3,1)*fd + D(3,2)*ff + D(3,3)*fkd)*w0;
 pfkq= -Rkq*(Q(2,1)*fq + Q(2,2)*fkq)*w0;
 ppang=(1/J)*((w0^2/(4))*(fd*fq*(Q(1,1)-D(1,1))+...
 Q(1,2)*fd*fkq-D(1,2)*fq*ff-D(1,3)*fq*fkd)-Tm-Kd*pang);
 pang= ang;

 sys=[pfd;pfq;pff;pfkd;pfkq;ppang;pang];

case 3,
 fd=x(1);fq=x(2);ff=x(3);fkd=x(4);fkq=x(5);
 pang=x(6);ang=x(7);
 % DQ transformation
 ud  =(2*Vb/3)*(cos(ang)+cos(ang-2*pi/3)+cos(ang-4*pi/3));
 uq  =(2*Vb/3)*(sin(ang)+sin(ang-2*pi/3)+sin(ang-4*pi/3));
 Id  = w0*D*[fd;ff;fkd];
 Iq  = w0*Q*[fq;fkq];
 id  = Id(1);
 iq  = Iq(1);
 It  = sqrt(id^2 + iq^2);
 Vt  = sqrt(ud^2 + uq^2);
 P   = ud*id + uq*iq;
 Q1  = ud*iq - uq*id;
 sys =[P;Q1;Vt;It];

case { 2, 4, 9 }
 sys = []; % Unused flags
otherwise
 error(['Unhandled flag = ',num2str(flag)]);
end
```

F.10 SM Matrix Initialisation file

```
%*******************************************
%* PROGRAM "igen_init"                     *
%* Initialisation file for igen            *
%*                                         *
%* Written by   Paramasivan Chetty         *
%*              Natal University           *
%*              09 September 2000          *
%*                                         *
%*******************************************
global inv_X RR WR wb;
%-------------------------------------------
%OPERATING POINT
Vb      = 1.047;
Pb      = 0.8;
Qb      = 0.4;
Rt      = (0.0071 + 0.01425)*0;
Xt      = (0.1295 + 0.3755)*0;

%-------------------------------------------
%GENERATOR MATRICES

%inductances
X=[-(Lmq+Ll)0 0 0 Lmq;
   0 -(Lmd+Ll) Lmd Lmd 0;
   0 -Lmd (Lmd+Llfd) Lmd 0;
   0 -Lmd Lmd (Lmd+Llkd) 0;
   -Lmq 0 0 0 (Lmq+Llkq)];

inv_X = inv(X);

%resistances
r     = [Rs Rs -Rfd -Rkd -Rkq];
RR    = diag(r);

%rotational matrix
WR=[0 (Lmd+Ll) -Lmd -Lmd 0;
   -(Lmq+Ll) 0 0 0 Lmq;
   0 0 0 0 0;
   0 0 0 0 0;
   0 0 0 0 0];

%-------------------------------------------
```

```
%CALCULATIONS
phi = angle(Pb+j*Qb);
I   = sqrt(Pb^2+Qb^2)/Vb;
tho = atan((I*(Xq+Xt)*cos(phi)-I*(Rs+Rt)*sin(phi))/...
          (Vb+I*(Rs+Rt)*cos(phi)+I*(Xq+Xt)*sin(phi)));
gam = pi/2 - (tho + phi);
ido = I*cos(gam);
iqo = I*sin(gam);
vbd = Vb*sin(tho);
vbq = Vb*cos(tho);
vtd = vbd - iqo*Xt + ido*Rt;
vtq = vbq + iqo*Rt + ido*Xt;
Vt  = sqrt(vtd^2 + vtq^2);
Uod = vtd - iqo*Xq + ido*Rs;
Uoq = vtq + ido*Xd + iqo*Rs;
Uo  = sqrt(Uod^2 + Uoq^2);
ifdo= Uo/Xmd;
Pm  = vtd*ido + vtq*iqo;
Qm  = vtd*iqo - vtq*ido;
dwo = 0;
```

F.11 SM Matrix Execution file

```
%*******************************************
%* PROGRAM "igen_sfun"                      *
%* S-Function file for igen                 *
%*                                          *
%* Written by   Paramasivan Chetty          *
%*              Natal University            *
%*              5 October 2000              *
%*                                          *
%*******************************************
function [out_vector] = igen_sfun(in_vector);
%-----------------------------------------------
% GLOBALS
global RR inv_X WR wb;
global Lmd Lmq Ll H;

%-----------------------------------------------
% CALCULATIONS
i     = in_vector(1:5);
v     = in_vector(6:10);
Tm    = in_vector(11);
wr    = in_vector(12);

p_i   = inv_X*wb*(v + (RR + (wr+1)*WR)*i);   % (wr+1)

phiq = -(Lmq + Ll)*i(1) + Lmq*i(5);
phid = -(Lmd + Ll)*i(2) + Lmd*(i(3) + i(4));

Te   = phid*i(1) - phiq*i(2);
pwr  = (1/2/H)*(Tm - Te);

out_vector  = [p_i' pwr];
```

F.12 Krause Machine Initialisation file

```
%******************************************
%* PROGRAM "krause_init"                 *
%* Initialisation file for krause_gen     *
%*                                        *
%* Written by   Paramasivan Chetty        *
%*              Natal University          *
%*              09 September 2000         *
%*                                        *
%******************************************
%------------------------------------------
%INITIALISATION
%GLOBALS
global phido phiqo phifdo phikdo phikqo dwo tho;
global Ll Lmd Lmq Ld Lq Lfd Lkd Lkq Lad Laq Rs...
                                 Rfd Rkd Rkq web;
global Llfd Llkd Llkq;
global H Kd;
global vdo vqo Pm iqo ido;

%------------------------------------------
%OPERATING POINT
Vb      = 1.0;
Pb      = 0.3;
Qb      = 0.5;
Rt      = 0;
Xt      = 0;

%------------------------------------------
%NOMINAL VALUES
fn      = 50;       %frequency of operation
web     = 2*pi*fn;  %electrical freq in rad/sec

%------------------------------------------
%ELECTRICAL VARIABLES
Xl      = 0.11;    %p.u
Xmd     = 1.98;    %p.u
Xmq     = 1.87;    %p.u
Xd      = Xmd+Xl;  %p.u
Xq      = Xmq+Xl;  %p.u
Rs      = 0.006;   %p.u
Xf      = 0.1;     %p.u
Rfd     = 0.00099; %p.u
```

```
Xkd     = 0.125;   %p.u
Rkd     = 0.0212;  %p.u
Xkq     = 0.257;   %p.u
Rkq     = 0.027;   %p.u
%In p.u, inductances are equal to reactances. (cf.Kundur)
Ll      = Xl;
Lmd     = Xmd;
Lmq     = Xmq;
Llfd    = Xf;
Llkd    = Xkd;
Llkq    = Xkq;
% Misc. inductances
Ld      = Ll+Lmd;
Lq      = Ll+Lmq;
Lfd     = Llfd+Lmd;
L1      = Lmq*(Ll+Llkq)+Ll*Llkq;
L2      = (Lmd*Ll*(Llfd+Llkd))+(Llfd*Llkd*(Ll+Lmd));
Lad     = Lmd*Ll*Llfd*Llkd/L2;
Laq     = Lmq*Ll*Llkq/L1;

%-------------------------------------------
% MECHANICAL VARIABLES
H       = 5.6;      %inertia constant in seconds
Kd      = 0;        %damping constant

%---------------------------------------
%CALCULATIONS
phi = angle(Pb+j*Qb); % power angle
I   = (sqrt(Pb^2+Qb^2))/Vb; % terminal current
Ia  = I*cos(-phi)+j*I*sin(-phi); % calculate complex I
Vt  = Vb;
% calculate complex excitation voltage
U   = Vt + (Rs + j*Xd)*Ia;
tho = angle(U); % calculate rotor angle;
vdo = Vb*sin(tho);
vqo = Vb*cos(tho);
Uo  = abs(U); % calculate magnitude of excitation V
pwr_ang = phi*180/pi; % radians to degrees
Pm  = (abs(Vb)*abs(Uo)/Xd)*sin(tho);% Pm calculation
i   = sqrt(2)*abs(Ia)*cos(-phi);

%---------------------------------------
%INITIAL VALUES
dwo     = 0;              %init. speed deviation
```

```
iao     = I;
ibo     = I;
ico     = I;               %init. current amplitudes
phao    = 90 - pwr_ang;
phbo    = -30 - pwr_ang;
phco    = 210 - pwr_ang;   %init. current angles

%---------------------------------------------
%FIELD VALUES
Vfopu   = Uo;              %init. field voltage (pu)
%nominal field voltage viewed from stator
Vfnp    = Rfd/Lmd;
Vfo     = Vfopu*(Rfd/Lmd); %initial field voltage

%-----------------------------------------
% Conversion of initial current phasors
% to instantaneous values.
[scrap,iao] = pol2cart(phao*pi/180,iao);
[scrap,ibo] = pol2cart(phbo*pi/180,ibo);
[scrap,ico] = pol2cart(phco*pi/180,ico);

%-----------------------------------------
% Initial id, iq and ifd current components
ifdo    = Vfo/Rfd;
i2      = [iao ibo ico]';
dpt     = 2*pi/3;
iqo     = 2/3*[cos(tho) cos(tho-dpt) cos(tho+dpt)]*i2;
ido     = 2/3*[sin(tho) sin(tho-dpt) sin(tho+dpt)]*i2;

%-----------------------------------------
% Initial fluxes
phiqo   = -Lq*iqo;
phikqo  = -Lmq*iqo;
phido   = (-Ld*ido+Lmd*ifdo);
phikdo  = (Lmd*(-ido+ifdo));
phifdo  = (Lfd*ifdo+Lmd*(-ido));

%-----------------------------------------
%DISPLAY
clc;
I       = sprintf('tho=%d,iqo=%d,ido=%d',tho,iqo,ido);
V       = sprintf('vqo=%d,vdo=%d,Uo=%d',vqo,vdo,Uo);
phiQ    = sprintf('phiqo=%d,phikqo=%d,Vfo=...
                      %d',phiqo,phikqo,Vfo);
```

```
phiD    = sprintf('phido=%d,phifdo=%d,phikdo=...
                        %d',phido,phifdo,phikdo);
disp(I);
disp(V);
disp(phiQ);
disp(phiD);
```

F.13 Krause Machine Execution file

```
%******************************************
%* PROGRAM "krause_sfun"                  *
%* S-Function file for krause_gen         *
%*                                        *
%* Written by   Paramasivan Chetty        *
%*              Natal University          *
%*              09 September 2000         *
%*                                        *
%******************************************
function [sys,x0,str,ts] = krause_sfun(t,x,u,flag)
%------------------------------------------
% GLOBALS
global wb H Xad Xaq Xa Xf Xmd Xkd Xkq Ra Rf Rkd Rkq;

%------------------------------------------
% DEFINE VARIABLES
wb  = 2*pi*60;
we  = 1.0;
P   = 4;
J   = 35.1e6;
Xmd = (0.850-0.120)/we;
Xmq = (0.480-0.120)/we;
Xa  = (0.120)/we;
Ra  = 0.0019;
Xf  = 0.2049/we;
Rf  = 0.00041;
Xkd = 0.160/we;
Rkd = 0.0141;
Xkq = 0.1029/we;
Rkq = 0.0136;
Xaq = 1/(1/Xmq + 1/Xa + 1/Xkq);
Xad = 1/(1/Xmd + 1/Xa + 1/Xf + 1/Xkd);

%------------------------------------------
%DEFINE BASE QUANTITIES
VB  = sqrt(2/3)*20e3; % V = rms voltage per phase
PB  = 325e6*0.85;
IB  = (2/3)*PB/VB;
ZB  = VB/IB;
TB  = (PB*P)/(2*wb);
H   = 0.5*(2/P)^2*J*wb^2/PB;
```

```
%-------------------------------------------
% BEGIN
switch flag,

  case 0,
    sys =[7 0 4 4 0 0 1];
    x0  =[0 0 0 0 0 0 0];
    str =[];
    ts  =[0 0];

 case 1,
    fd=x(1);fq=x(2);ff=x(3);fkd=x(4);fkq=x(5);
    wr=x(6);ang=x(7);
    vf=u(1);Tm=u(2);vd=u(3);vq=u(4);

    %DIFFERENTIAL EQUATIONS
    pfd = wb*(vd + (wr/wb)*fq + (Ra/Xa)*(Xad*(fd/Xa +...
                                  ff/Xf + fkd/Xkd)-fd));
    pfq = wb*(vq - (wr/wb)*fd + (Ra/Xa)*(Xaq*(fq/Xa +...
                                          fkq/Xkq)-fq));
    pff = wb*((Rf/Xmd)*vf + (Rf/Xf)*(Xad*(fd/Xa +...
                              ff/Xf + fkd/Xkd)-ff));
    pfkd= wb*((Rkd/Xkd)*(Xad*(fd/Xa + ff/Xf +...
                                 fkd/Xkd)-fkd));
    pfkq= wb*((Rkq/Xkq)*(Xaq*(fq/Xa + fkq/Xkq)-fkq));

    %sidebar
    iq  = -((1/Xa)*(fq - Xaq*(fq/Xa + fkq/Xkq)));
    id  = -((1/Xa)*(fd - Xad*(fd/Xa + fkd/Xkd + ff/Xf)));
    Te  = fd*iq - fq*id;

    %DIFFERENTIAL EQUATIONS
    pwr = (wb/2*H)*(Tm - Te);
    pang= wr;

    %INTEGRATE
    sys=[pfd pfq pff pfkd pfkq pwr pang];

case 3,
   fd=x(1);fq=x(2);ff=x(3);fkd=x(4);fkq=x(5);
   wr=x(6);ang=x(7);
   iq  = -((1/Xa)*(fq - Xaq*(fq/Xa + fkq/Xkq)));
   id  = -((1/Xa)*(fd - Xad*(fd/Xa + fkd/Xkd + ff/Xf)));
   sys=[id iq wr ang];
 case { 2, 4, 9 }
     sys = []; % Unused flags
  otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
```

F.14 Initial Conditions Solver file

```
%****************************************
%*  Single Machine Infinite Bus System    *
%*  Initial Conditions Solver             *
%*  ----------------------------------    *
%*                                        *
%*  Inputs: Real power Pt                 *
%*          Reactive power Qt             *
%*          Terminal voltage Et           *
%*                                        *
%*                                        *
%*  Outputs: Shaft speed                  *
%*           Rotor angle                  *
%*           Bus voltage                  *
%*           A, B, C and D matrices       *
%*                                        *
%*  ------------Revision history--------- *
%*  created on: 15 June 1999 by P Chetty  *
%* =============                          *
%*  last modified:  16 March 2000         *
%*****************************************
function [] = ics(Et,Pt,Qt)
%-------------------------------------------
% INITIALISATION
%Et = 1.0;           % Bus voltage
%Pt = 0.8;
%Qt = 0.4;
Xd = 2.09;
Xq = 1.98;
Xl = 0.11;
Ra = 0.006;
Re = 0.0;
Xe = 0.0;

clc;
msg=sprintf('Steady state values with...
Et=%4.4f p.u, Pt=%4.4f p.u and Qt=%4.4f p.u',Et,Pt,Qt);
disp(msg);
disp(' ');

%-------------------------------------------
% Calculation of parameters
Xadu=Xd-Xl;
```

```
Xaqu=Xq-Xl;
%It=sqrt(Pt^2+Qt^2)/Et; %Kundur pg.101
It=conj(Pt+j*Qt)/conj(Et); %Kundur pg.733
msg=sprintf('Terminal current=%4.4f p.u...
%4.4f amps(7.87)',abs(It),It*7.87);
disp(msg)

%-------------------------------------------
% Calculation of total saturation factors
Asat=0.019;
Bsat=2.48;
Ea=Et+(Ra+j*Xl)*It;
Ea=abs(Ea);
msg=sprintf('Internal voltage=%4.4fp.u...
%4.4f volts(127)',Ea,Ea*127);
disp(msg)
Fat=Ea;
FTI=0.83; % Calculated from the saturation graph.
FI=Asat*exp(Bsat*(Ea-FTI));
Ksd=Fat/(Fat+FI);
Ksq=Ksd;

%-------------------------------------------
% Cont. of parameter calculation
Xds=Ksd*Xadu+Xl;
Xqs=Ksq*Xaqu+Xl;
Xqs=Xq;
Xds=Xd;
% acos returns an ambiguous result depending
% on the quadrant
%pwr_ang=acos(Pt/(Et*It));
pwr_ang=angle(Pt+j*Qt);
msg=sprintf('Power angle=%4.4f deg',pwr_ang*(180/pi));
disp(msg)
%Use this equation to prevent ambiguity for atan function.
int_ang=angle(Et+(Ra+j*Xqs)*It);
msg=sprintf('Internal angle=%4.4f deg',...
int_ang*(180/pi));
disp(msg)
ed0=Et*sin(int_ang);
eq0=Et*cos(int_ang);
id0=It*sin(int_ang+pwr_ang);
iq0=It*cos(int_ang+pwr_ang);
Ebd0=ed0-Re*id0+Xe*iq0;
```

```
Ebq0=eq0-Re*iq0-Xe*id0;
delta0=atan2(Ebd0,Ebq0); %Use the advanced atan2 function.
msg=sprintf('Rotor angle=%4.4f deg',...
delta0*(180/pi));
disp(msg)

Eb=(Ebd0^2+Ebq0^2)^0.5;
msg=sprintf('Terminal voltage=%4.4f p.u...
%4.4f volts(127)',Eb,Eb*127);
disp(msg)
% assumption that ifd0, Fad0 and Faq0 is calculated
% using total saturation. Kundur p.746
Xads=Xadu*Ksd;
Xaqs=Xaqu*Ksq;
ifd0=(eq0+Ra*iq0+Xds*id0)/Xads;
msg=sprintf('Field current=%4.8f p.u...
%4.4f amps(1.218)',ifd0,ifd0*1.218);
disp(msg)
Efd0=Xadu*ifd0;
msg=sprintf('Field voltage=%4.4f p.u
%4.4f volts(1231)',Efd0,Efd0*1231);
disp(msg)

%-------------------------------------------
% dq transformation of id, iq currents
theta = pi/2 - int_ang + pwr_ang;
mf=[cos(theta),sin(theta),1; ...
    cos(theta-2*pi/3),sin(theta-2*pi/3),1; ...
    cos(theta-4*pi/3),sin(theta-4*pi/3),1];
Idqz=[id0;iq0;0]; % assuming balanced conditions
Iabc=mf*Idqz;
Ia=Iabc(1);
Ib=Iabc(2);
Ic=Iabc(3);
wIa=sprintf('Line current Ia=%f%f amps(7.87)',Ia,Ia*7.87);
wIb=sprintf('Line current Ib=%f%f amps(7.87)',Ib,Ib*7.87);
wIc=sprintf('Line current Ic=%f%f amps(7.87)',Ic,Ic*7.87);
disp(wIa);
disp(wIb);
disp(wIc);
disp(' ');
```

F.15 Linearised PSB Model file

```
%*****************************************
%* PROGRAM "lingen"                      *
%* Linearisation of the PSB equations.   *
%* This m-file forms the ABCD matrices.  *
%*                                       *
%* Written by P Chetty                   *
%* University of Natal                   *
%* 13 September 2000                     *
%*                                       *
%*****************************************
function[Out_vector] = lingen(In_vector);
%-----------------------------------------
% GLOBALS
global phido phiqo phifdo phikdo phikqo dwo tho;
global Ll Lmd Lmq Ld Lq Lfd Lkd Lkq Lad Laq Rs...
                                Rfd Rkd Rkq web;
global Llfd Llkd Llkq;
global H Kd;
global vdo vqo Pm;

%-----------------------------------------
%DEFINE VARIABLES
wo  = 0;

%-----------------------------------------
%delta pphid
U1  = web*0;                      %delta vd
FD1 = (Rs*web/Ll)*(Lad/Ll - 1);   %delta phid
FD2 = web + web*wo;               %delta phiq
FD3 = Rs*Lad*web/Llfd/Ll;         %delta phifd
FD4 = Rs*Lad*web/Llkd/Ll;         %delta phikd
FD5 = 0;                          %delta phikq
FD6 = web*phiqo;                  %delta wr
FD7 = vqo*web;                    %delta theta

%delta pphiq
U2  = web*0;                      %delta vq
FQ1 = -web-web*wo;                %delta phid
FQ2 = (Rs*web/Ll)*(Laq/Ll - 1);   %delta phiq
FQ3 = 0;                          %delta phifd
FQ4 = 0;                          %delta phikd
FQ5 = Rs*Laq*web/Llkq/Ll;         %delta phikq
```

```
FQ6 = -web*phido;                 %delta wr
FQ7 = -vdo*web;                   %delta theta

%delta pphifd
U3  = web*Rfd/Lmd;                %delta vfd
FF1 = Rfd*Lad*web/Llfd/Ll;        %delta phid
FF2 = 0;                          %delta phiq
FF3 = (Rfd*web/Llfd)*...
(Lad/Llfd - 1);                   %delta phifd
FF4 = Rfd*Lad*web/Llfd/Llkd;      %delta phikd
FF5 = 0;                          %delta phikq
FF6 = 0;                          %delta wr
FF7 = 0;                          %delta theta

%delta pphikd
FKD1= Rkd*Lad*web/Llkd/Ll;        %delta phid
FKD2= 0;                          %delta phiq
FKD3= Rkd*Lad*web/Llkd/Llfd;      %delta phifd
FKD4= (Rkd*web/Llkd)*...
(Lad/Llkd - 1);                   %delta phikd
FKD5= 0;                          %delta phikq
FKD6= 0;                          %delta wr
FKD7= 0;                          %delta theta

%delta pphikq
FKQ1= 0;                          %delta phid
FKQ2= Rkq*Laq*web/Llkq/Ll;        %delta phiq
FKQ3= 0;                          %delta phifd
FKQ4= 0;                          %delta phikd
FKQ5= (Rkq*web/Llkq)*...
(Laq/Llkq - 1);                   %delta pphikq
FKQ6= 0;                          %delta wr
FKQ7= 0;                          %delta theta

%delta Te
T1  = (phiqo/Ll^2)*...
(Laq-Lad)+Laq*phikqo/Llkq/Ll;     %delta phid
T2  = (phido/Ll^2)*...
(Laq-Lad)-(Lad/Ll)*...
(phifdo/Llfd-phikdo/Llkd);        %delta phiq
T3  = -Lad*phiqo/Llfd/Ll;         %delta phifd
T4  = -Lad*phiqo/Llkd/Ll;         %delta phikd
T5  = Laq*phido/Llkq/Ll;          %delta pphikq
```

```
%delta pwr
U4  = 1/(2*H + 2*H*wo);             %delta pm
W1  = -T1/2/H;                      %delta phid
W2  = -T2/2/H;                      %delta phiq
W3  = -T3/2/H;                      %delta phifd
W4  = -T4/2/H;                      %delta phikd
W5  = -T5/2/H;                      %delta phikq
W6  = Pm/(2*H*(1+wo)^2)-Kd;         %delta wr
W7  = 0;                            %delta theta

%delta theta
TH1 = 0;                            %delta phid
TH2 = 0;                            %delta phiq
TH3 = 0;                            %delta phifd
TH4 = 0;                            %delta phikd
TH5 = 0;                            %delta phikq
TH6 = web;                          %delta wr
TH7 = 0;                            %delta theta

%-------------------------------------------
%ABCD MATRICES

%form A
A=[FD1  FD2  FD3  FD4  FD5  FD6  FD7;
   FQ1  FQ2  FQ3  FQ4  FQ5  FQ6  FQ7;
   FF1  FF2  FF3  FF4  FF5  FF6  FF7;
   FKD1 FKD2 FKD3 FKD4 FKD5 FKD6 FKD7;
   FKQ1 FKQ2 FKQ3 FKQ4 FKQ5 FKQ6 FKQ7;
   W1   W2   W3   W4   W5   W6   W7;
   TH1  TH2  TH3  TH4  TH5  TH6  TH7];

%form B
% vdo       vqo     vfd     pm
B=[U1       0       0       0;
   0        U2      0       0;
   0        0       U3      0;
   0        0       0       0;
   0        0       0       0;
   0        0       0       U4;
   0        0       0       0];

%form delta id,iq
ID1 = -1/L1 + Lad/L1^2; %delta phid
ID2 = 0;                %delta phiq
```

```
ID3 = Lad/Llfd/Ll;        %delta phifd
ID4 = Lad/Llkd/Ll;        %delta phikd
ID5 = 0;                  %delta phikq
ID6 = 0;                  %delta wr
ID7 = 0;                  %delta theta

IQ1 = 0;                  %delta phid
IQ2 = -1/Ll + Laq/Ll^2;   %delta phiq
IQ3 = 0;                  %delta phifd
IQ4 = 0;                  %delta phikd
IQ5 = Laq/Llkq/Ll;        %delta phikq
IQ6 = 0;                  %delta wr
IQ7 = 0;                  %delta theta

%form delta vd,vq
VD1 = 0;                  %delta phid
VD2 = 0;                  %delta phiq
VD3 = 0;                  %delta phifd
VD4 = 0;                  %delta phikd
VD5 = 0;                  %delta phikq
VD6 = 0;                  %delta wr
VD7 = vqo;                %delta theta

VQ1 = 0;                  %delta phid
VQ2 = 0;                  %delta phiq
VQ3 = 0;                  %delta phifd
VQ4 = 0;                  %delta phikd
VQ5 = 0;                  %delta phikq
VQ6 = 0;                  %delta wr
VQ7 = vdo;                %delta theta


%form C
C=[IQ1  IQ2 IQ3 IQ4 IQ5 IQ6 IQ7; %delta iq
   ID1  ID2 ID3 ID4 ID5 ID6 ID7; %delta id
   VQ1  VQ2 VQ3 VQ4 VQ5 VQ6 VQ7; %delta vq
   VD1  VD2 VD3 VD4 VD5 VD6 VD7; %delta vd
   0    0    0  0   0   0    1]; %delta theta

%form D
D1  = [1 0 0 0; 0 1 0 0];
D   = [D1;zeros(3,4)];

%-------------------------------------------
```

```
%CALCULATIONS
phid     = In_vector(1);
phiq     = In_vector(2);
phifd    = In_vector(3);
phikd    = In_vector(4);
phikq    = In_vector(5);
wr       = In_vector(6);
th       = In_vector(7);
vd       = In_vector(8);
vq       = In_vector(9);
vfd      = In_vector(10);
pm       = In_vector(11);

x        = [phid phiq phifd phikd phikq wr th]';
u        = [vd vq vfd pm]';
xdot     = A*x + B*u;
y        = C*x + D*u;

%-------------------------------------------
%OUTPUT
Out_vector = xdot;
```

F.16 Distributed Shaft Dynamics file

```
%******************************************
%* PROGRAM "shaftdyn_sfun"                *
%* S-Function file for shaft dynamics     *
%*                                        *
%* Written by   Paramasivan Chetty        *
%*              Natal University          *
%*              15 September 2000         *
%*                                        *
%******************************************
function [sys,x0,str,ts] = shaftdyn_sfun(t,x,u,flag)
%------------------------------------------
% definition of ABCD
J1 = 2.355e-3;
J2 = 7.608e-3;
J3 = 7.576e-3;
J4 = 7.779e-3;
J5 = 1.063e-2;
J6 = 2.192e-5;

K12 = 10.63;
K23 = 25.34;
K34 = 23.42;
K45 = 26.93;
K56 = 70.53;

D1 = 7.4e-4;
D2 = 2.39e-3;
D3 = 2.38e-3;
D4 = 2.44e-3;
D5 = 3.34e-3;
D6 = 6.888e-5;

AA = [0 1 0 0 0 0 0 0 0 0 0 0;
   -K12/J1 -D1/J1 K12/J1 0 0 0 0 0 0 0 0 0;
   0 0 0 1 0 0 0 0 0 0 0 0;
   K12/J2 0 -(K12+K23)/J2 -D2/J2 K23/J2 0 0 0 0 0 0 0;
   0 0 0 0 0 1 0 0 0 0 0 0;
   0 0 K23/J3 0 (K23+K34)/J3 -D3/J3 K34/J3 0 0 0 0 0;
   0 0 0 0 0 0 0 1 0 0 0 0;
   0 0 0 0 K34/J4 0 -(K34+K45)/J4 -D4/J4 K45/J4 0 0 0;
   0 0 0 0 0 0 0 0 0 1 0 0;
   0 0 0 0 0 0 K45/J5 0 -(K45+K56)/J5 -D5/J5 K56/J5 0;
   0 0 0 0 0 0 0 0 0 0 0 1;
   0 0 0 0 0 0 0 0 K56/J6 0 -K56/J6 -D6/J6];

B = [1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0];
C = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,1];
D = [0];

%------------------------------------------
% Dispatch the flag.
switch flag,
case 0
% Initialisation
[sys,x0,str,ts]=mdlInitializeSizes(AA,B,C,D);
case 1
% Calculate derivatives
sys = mdlDerivatives(t,x,u,AA,B,C,D);
case 3
% Calculate outputs
sys = mdlOutputs(t,x,u,AA,B,C,D);
case { 2, 4, 9 } % Unused flags
sys = [];
otherwise
% Error handling
error(['Unhandled flag = ',num2str(flag)]);
end

%------------------------------------------
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times
% for the S-function.
function [sys,x0,str,ts] = mdlInitializeSizes(AA,B,C,D)
% Call simsizes for a sizes structure, fill it in and
% convert it to a sizes array.
sizes = simsizes;
sizes.NumContStates = 12;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 1;
sizes.NumInputs = 1;
sizes.DirFeedthrough = 0; % Matrix D is nonempty.
sizes.NumSampleTimes = 1;
sys = simsizes(sizes);
% Initialize the initial conditions.
x0 = zeros(12,1);
% str is an empty matrix.
```

```
str = [];
% Initialize the array of sample times; in this example
% the sample time is continuous, so set ts to 0 and its
% offset to 0.
ts = [0 0];

%------------------------------------------
% mdlDerivatives
% Return the derivatives for the continuous states.
function sys = mdlDerivatives(t,x,u,AA,B,C,D)
sys = AA*x + B*u;

%------------------------------------------
% mdlOutputs
% Return the block outputs.
function sys = mdlOutputs(t,x,u,AA,B,C,D)
sys = C*x + D*u;
```

F.17 FFT Calculation file

```
%*****************************************
%* PROGRAM "fft"                         *
%* M-file for calculating FFT from data. *
%*                                       *
%* Written by   Paramasivan Chetty       *
%*              Natal University         *
%*              09 May 2002              *
%*                                       *
%*****************************************
%---------------------------------------------
% INITIALISATION
X=load('fr\f1.txt');  %load fft data file here
freq=0.2;
x1=X(:,3);
x2=X(:,2);

Fs=10;                % sampling frequency
Fn=Fs/2;              % Nyquist frequency

%---------------------------------------------
% CALCULATIONS
NFFT=2.^(ceil(log(length(x1))/...
log(2)));             % Next highest power of 2
                      % greater than length(x).

FFTX1=fft(x1,NFFT);   % Take FFT, padding with zeros.
FFTX2=fft(x2,NFFT);   % Take FFT, padding with zeros.

P1=(angle(FFTX1))*(180/pi); % Calculate phase
P2=(angle(FFTX2))*(180/pi); % Calculate phase
fp=(0:length(P1)-1)'/length(P1)*Fs; % frequency vector
                                    % length(FFTX)==NFFT
NumUniquePts = ceil((NFFT+1)/2);
FFTX1=FFTX1(1:NumUniquePts); % FFT is symmetric,throw away
FFTX2=FFTX2(1:NumUniquePts); % FFT is symmetric,throw away

                      % second half
MX1=abs(FFTX1);       % Take magnitude of X
MX1=MX1*2;            % Multiply by 2 to take into
MX2=abs(FFTX2);       % account the fact that we
MX2=MX2*2;            % threw out second half of
                      % FFTX above
```

```
MX1(1)=MX1(1)/2;        % Account for endpoint
MX2(1)=MX2(1)/2;        % uniqueness
MX1(length(MX1))=MX1(length(MX1))/2; % NFFT is even
MX1=MX1/length(x1);     % Scale the FFT so that it is
                        % not a function of the length of x.
MX2(length(MX2))=MX2(length(MX2))/2;
MX2=MX2/length(x2);
f=(0:NumUniquePts-1)*2*Fn/NFFT;
MY1=interp1(f,MX1,freq)
MY2=interp1(f,MX2,freq)
20*log10(MY2/MY1)
PY1=interp1(fp,P1,freq)
PY2=interp1(fp,P2,freq)
PY2-PY1

%------------------------------------------
% PLOT RESULT
figure;
subplot(2,1,1);
plot(f, MX1);grid on;
axis([0,2,0,0.03]);
title('FFT of V_{ref} at 0.2Hz');
xlabel('Frequency (Hz)');
ylabel('Voltage (V)');
subplot(2,1,2);
plot(f, MX2);grid on;
axis([0,2,0,0.1]);
title('FFT of \Delta\omega at 0.2Hz');
xlabel('Frequency (Hz)');
ylabel('\Delta\omega (V)');

figure;
subplot(2,1,1);
plot(fp, P1);grid on;
axis([0,2,-200,200]);
title('FFT of V_{ref} at 0.2Hz');
xlabel('Frequency (Hz)');
ylabel('Phase (deg)');
subplot(2,1,2);
plot(fp, P2);grid on;
axis([0,2,-200,200]);
title('FFT of \Delta\omega at 0.2Hz');
xlabel('Frequency (Hz)');
ylabel('\Delta\omega (deg)');
```