Design of an Adaptive Power System Stabilizer

by

Gregory Anthony Jackson

A Thesis submitted to the Faculty of Graduate Studies in
partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering.

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba; Canada

*I would like to dedicate this thesis to my mother,
a woman who often cooked for me over the course
of my studies and who was always supportive.
Thank-you for the food.*

Table of Contents

Table of Contents III

List of Tables V

List of Figures VI

Glossary VIII

Abstract IX

Acknowledgements X

1 Introduction 1

1.1 OVERVIEW 1
1.2 BACKGROUND 3
1.2.1 Algorithm Selection 3
1.2.2 Recursive Least-Squares (RLS) Algorithm 5
1.2.3 Generalized Predictive Control (GPC) Algorithm 8
1.2.4 Continuous versus Discrete Control 13
1.2.5 Calculation/Simulation Platform 16
2 System Identification 17

2.1 SIMULATION CONSIDERATIONS 17
2.1.1 Identification Target 17
2.1.2 Sampling Rate 18
2.1.3 Dither Signal 18
2.1.4 RLS Startup 20
2.1.5 Parameter Settling and Buffering 21
2.1.6 General Investigative Approach 23
2.1.7 Experimental/Simulation Setup 25
2.2 SIGNAL QUALITY 28
2.2.1 Simulation Results 28
2.2.2 Discussion and Analysis 31
2.3 MODEL ADAPTABILITY 35
2.3.1 Adaptability 35
2.3.2 Stability 37
2.3.3 Discussion 38
2.4 MODEL ORDER SELECTION 39
2.4.1 Simulation Results 40
2.4.2 Discussion 42
2.5 SUMMARY 43
3 Controller Synthesis 46

3.1 BASIC CONTROLLER 47
3.2 CONTROLLER COMPLEXITY 50
3.2.1 Process Model Simplification 50
3.2.2 Reduced-Order GPC vs Full-Order GPC 52
3.2.3 Discussion 57
3.3 CONTROLLER PARAMETERS 57
3.3.1 General Observations 57
3.3.2 Control Parameter Selection 61
3.4 CONTROLLER OFFSET ELIMINATION 64
3.4.1 Composite Input Signal 65
3.4.2 Integrator Removal 70
3.4.3 Lossy Integrator 72
3.4.4 Discussion 75
3.5 GPC PSS VS TRADITIONAL PSS 77
3.5.1 Nominal Operating Point Performance 78
3.5.2 Off-Nominal Operating Point Performance 83
3.6 SUMMARY 86
4 Adaptive Controller 89

4.1 VIABILITY OF AN ESTIMATED-MODEL BASED GPC PSS 89
4.1.1 Correct Model Order: $N_a = 4$, $N_b = 4$ and $N_c = 0$ 89
4.1.2 Under-modeling and Over-modeling 94
4.2 CONTROL PARAMETER SELECTION 95
4.2.1 Fixed Control Parameters: 95
4.2.2 Self-tuning Control Parameters 96
4.3 SUMMARY 98
5 Conclusion 100

6 Future Work 102

APPENDIX A 104

APPENDIX B 111

APPENDIX C 118

APPENDIX D 123

APPENDIX E 127

References 134

List of Tables

Table 2-1: Accuracy of estimated dominant closed-loop mode as a function of the quality of the signals used to construct the estimated model. Indicated is the dither signal level, the pre- and post-filter SNRs, the error in real and imaginary components for three different sets of buffered estimated parameters and the estimated dominant mode from the last set of buffered parameters. 30

Table 2-2: Average value of the covariance matrix trace as a function of forgetting factor. The stability of the algorithm for each choice of forgetting factor is indicated. 38

Table 2-3: Estimated open-loop poles as function of model order assumption. 40

Table 2-4: Error in the estimated dominant mode of the process. 40

Table 2-5: Error in the dominant estimated mode's damping and frequency of oscillation as a function of the model's assumed model order. 42

Table 3-1 : Evolution of five different unstable closed-loop modes as a function of the weighting constant C_2. 69

Table 3-2: Contribution of the AVR and GPC PSS as a function of the weighting constants to the total steady-state exciter reference signal. 70

Table 3-3: Dominant modes and their properties for closed-looped processes employing Traditional PSS #1 and GPC PSS #1 designs. 78

Table 3-4: Dominant modes and their properties for closed-looped processes employing Traditional PSS #2 and GPC PSS #2 designs. 80

Table 3-5: Dominant modes and their properties for closed-looped processes employing Traditional PSS #3 and GPC PSS #3 designs 82

Table 4-1: Percentage error in the estimated dominant mode and its properties as a function of the SNR of the signal used to construct the associated process model. 93

List of Figures

Figure 1-1: Block diagram of the basic components of an adaptive controller. 2

Figure 1-2: Diagram showing the stages involved in the design, analysis and comparison of traditional and GPC PSSs. 15

Figure 2-1: An example of a pseudo-random binary sequence (PRBS). 19

Figure 2-2: Demonstration of the effect that small and large initial covariance matrices have on the initial rate of parameter convergence.. 20

Figure 2-3: Time evolution of the RLS estimated parameters using the constant forgetting factor buffering scheme. 21

Figure 2-4: Trigger Pattern for Parameter Freezing 22

Figure 2-5: RLS Estimated Parameters using the variable forgetting factor scheme......................... 22

Figure 2-6: Block diagram representing different candidate setups for investigation of the RLS algorithm... 23

Figure 2-7: Block diagrams showing how selecting *filter 1* and *filter 2* the same is the best means of minimizing their impact on the estimated parameters. 26

Figure 2-8: Column A shows a comparison of the noisy and clean machine speed deviation signals resulting from application of PRBSs of varying amplitude. Column B shows how the filtered signals compare to the clean signals. The variance of the output noise is kept constant at 1e-11. 27

Figure 2-9: Comparison of the time responses from selected RLS estimated models. 31

Figure 2-10: Plots of the error in the estimated dominant mode's frequency and damping as a function of the SNR of the signal input to the RLS algorithm.. 33

Figure 2-11: RLS algorithm employing a *constant forgetting factor buffering scheme* and a forgetting factor of 0.990 tracks changes in the real and imaginary parts of the process' dominant pole.. 36

Figure 2-12: RLS algorithm employing a *constant forgetting factor buffering scheme* and a forgetting factor of 1.000 tracks changes in the real and imaginary parts of the process' dominant pole.. 36

Figure 2-13: RLS algorithm employing a *variable forgetting factor buffering scheme* and a forgetting factor of 0.990 tracks changes in the real and imaginary parts of the process' dominant pole... 37

Figure 2-14: Comparison of the time responses from several estimated models and the actual model when each is subjected to an identical disturbance... 41

Figure 3-1: Diagram indicating the calculation of a closed-loop transfer function. 49

Figure 3-2: Comparison of machine speed deviation and GPC control signals from closed-loop processes employing the full-order and reduced-order GPC controllers. *(Test System 1)* 53

Figure 3-3: Trajectories of the dominant closed-loop modes resulting from application of the reduced and the full-order GPC controllers to various off-nominal operating conditions. *(Test System 1)*.. 54

Figure 3-4: Comparison of machine speed deviation and the control signals from closed-loop processes employing the full-order and reduced-order GPC controllers. *(Test System 2)* 55

Figure 3-5: Trajectories of the dominant closed-loop modes resulting from application of the reduced and the full-order GPC controllers to various off-nominal operating conditions. *(Test System 2)*.. 56

Figure 3-6: Mappings of achievable closed-loop damping as a function of the output horizon, N, and the control horizon, N_u. (A) Full-Order GPC and (B) Reduced-Order GPC. 58

Figure 3-7: Damping trajectories of the dominant closed-loop mode which results from application of the full and reduced-order GPC controllers. (N = 50, N_u = 15)..................................... 58

Figure 3-8: Trajectories of closed-loop poles for a GPC controlled process and the trajectory of the damping for the closed-loop's dominant oscillatory mode. 63

Figure 3-9: Stability of the GPC controller as a function of the control penalty, α. 64

Figure 3-10: Time-domain responses of several composite-input GPC-controlled processes when the machine is subjected to a 0.1 pu step increase in its mechanical torque input. 67

Figure 3-11: Dominant closed-loop mode damping trajectories which result when different composite-input GPC PSSs are subjected to varying off-nominal operating conditions. 68

Figure 3-12: Trajectory of closed-loop poles resulting from application of GPC controller without an integrator. The damping of the dominant closed-loop mode is also shown. 71

Figure 3-13: Trajectory of the damping of the closed-looped system's dominant mode under varying off-nominal operating conditions. 72

Figure 3-14: Time-domain responses of several lossy-integrator GPC-controlled processes when the machine is subjected to a 0.1 pu step increase in its mechanical torque input. 74

Figure 3-15: Trajectory of the damping of the closed-looped system's dominant mode under varying off-nominal operating conditions. 75

Figure 3-16: Comparison of the time responses from the closed-loop systems employing Traditional PSS #1 and GPC PSS #1. The responses of the open-loop system are added for the sake of reference. 79

Figure 3-17: Comparison of the time responses from the closed-loop systems employing Traditional PSS #2 and GPC PSS #2. 81

Figure 3-18: Comparison of the time responses from the closed-loop systems employing Traditional PSS #3 and GPC PSS #3. 83

Figure 3-19: Comparison of damping provided by GPC PSS #1 and Traditional PSS #1 during off-nominal closed-loop operation. 84

Figure 3-20: Comparison of damping provided by GPC PSS #2 and Traditional PSS #2 during off-nominal closed-loop operation. 85

Figure 3-21: Comparison of damping provided by GPC PSS #3 and Traditional PSS #3 during off-nominal closed-loop operation. 86

Figure 4-1: Comparison of the damping trajectories for the dominant closed-loop mode that results from application of different estimated-model based GPCs. 90

Figure 4-2: Comparison of machine speed deviation and GPC control signals from an analytic-model based GPC PSS and three estimated-model based GPC PSSs. The SNR of the signal used to construct the model for each controller is indicated. 91

Figure 4-3: Plots comparing time response of different estimated process models to the same disturbance. 92

Figure 4-4: Off-nominal operation closed-loop damping provided by several estimated-model based GPC PSSs. 94

Figure 4-5: Closed-loop damping trajectories for off-nominal operation. 96

Figure 4-6: Diagram illustrating three principally different ways of calculating closed-loop poles... 97

Figure 4-7: Comparison of estimated and actual damping trajectories for different estimated-model based GPC PSSs. (N =50, N_u =15).. 98

Glossary

APSS	Adaptive Power System Stabilizer
FIR	Finite Impulse Response
GPC	Generalized Predictive Control(ler)
IO	Input-Output
LFO	Low Frequency Oscillation
LHS	Left-Hand Side
LTI	Linear Time-Invariant
MIMO	Multi-Input Multi-Output
OMIB	One Machine Infinite Bus
PRBS	Pseudo-Random Binary Sequence
PSS	Power System Stabilizer
RHS	Right-Hand Side
RLS	Recursive Least Squares
RMS	Root Mean Square
SISO	Single-Input Single-Output
SNR	Signal-to-Noise Ratio

Abstract

Modern power networks are being driven ever closer to both their physical and operational limits. As a result, control systems are being increasingly relied on to assure satisfactory system performance. Power system stabilizers (PSSs) are one example of such controllers. Their purpose is to increase system damping and they are typically designed using a model of the network that is valid during nominal operating conditions. The limitation of this design approach is that during off-nominal operating conditions, such as those triggered by daily load fluctuations, performance of the controller can degrade.

The research presented in this report attempts to evaluate the possibility of employing an adaptive PSS as a means of avoiding the performance degradation precipitated by off-nominal operation. Conceptually, an adaptive PSS would be capable of identifying changes in the network and then adjusting its parameters to ensure suitable damping of the identified network. This work begins with a detailed look at the identification algorithm employed followed by a similarly detailed examination of the control algorithm that was used. The results of these two investigations are then combined to allow for a preliminary assessment of the performance that could be expected from an adaptive PSS.

The results of this research suggest that an adaptive PSS is a possibility but further work is needed to confirm this finding. Testing using more complex network models must be carried out, details pertaining to control parameter tuning must be resolved and closed-loop time domain simulations using the adaptive PSS design remain to be performed.

Acknowledgements

First, I would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and RTDS Technologies Inc. for funding my work. I would also like to thank my advisor Dr. U.D. Annakkage for the guidance and direction he provided me throughout my research. Finally, I would like to thank all the individuals who have been part of the University of Manitoba's Power Group; professors, technical staff, post-graduate fellows and students alike for the sum of all the small things they've contributed to my education. I would especially like to recognize In Kwon Park and Xi Lin, two people whose conversation and advice assisted me greatly in my work.

1 Introduction

1.1 Overview

Power system stability is an issue of increasing concern as transmission lines and generators are pushed ever closer to both their operational and physical limits. Control systems are being relied on more heavily to assure satisfactory system performance. An example of such a control system is the Power System Stabilizer (PSS). Its purpose is to increase damping of low-frequency power oscillations. This objective is generally achieved through modulation of a generator's output power via manipulation of its field excitation.

PSS design has traditionally been based on some model that accounts for the generator and existing excitation controls in addition to some assumed model for the power system that is valid under nominal operating conditions. The limitation of this approach is that in the event of certain contingencies or even daily load fluctuations the assumed model deviates from the actual one. The more the actual operating point deviates from the nominal operating point the more ill-suited the original controller's design for the control task assigned to it. Such a situation can lead to degradation in the ability of the PSS to function effectively.

Advances in digital computation power have brought the concept of an adaptive controller into the realm of possibility. The concept of an adaptive controller is not a novel idea and it entails identification of the process in question followed by the synthesis of a controller tailored specifically to the identified process. The identification and controller synthesis components of an adaptive controller each possess their own unique challenges. The block diagram of figure 1-1 illustrates how the identification and control components of the design are largely independent while also showing how the two interface.

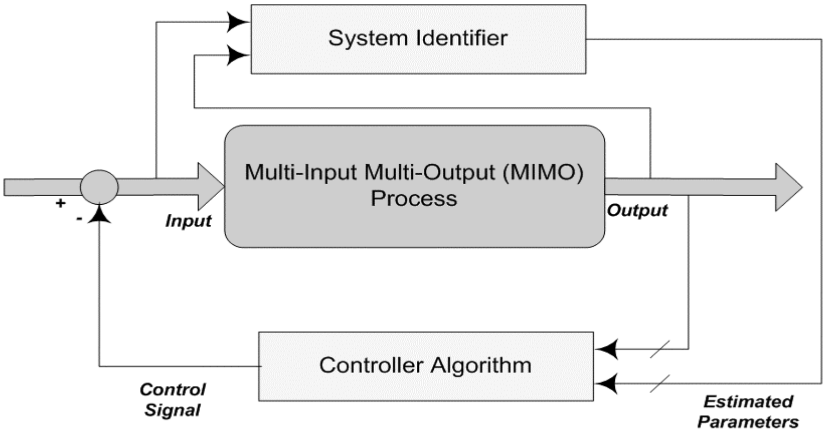

Figure 1-1: Block diagram of the basic components of an adaptive controller.

The process that will ultimately be controlled is in general a Multi-Input, Multi-Output (MIMO) system. The control algorithm is not unlike a regular controller that employs some feedback mechanisms. The difference here is that the controller requires, as an additional input, the identified model of the process being controlled. This input is then used in the derivation of an appropriate control signal. Despite generally involving processes of MIMO structure, most control problems revolve around knowledge of how a specific process output relates to a specific process input. This being the case, a Single-Input Single-Output (SISO) model of the relation of interest is usually sufficient to synthesize an effective controller.

The research presented in this work examines the practicality and potential effectiveness of applying an adaptive controller to damp Low Frequency Oscillations (LFOs). The result of such an application can be suitably described as an Adaptive

Power System Stabilizer (APSS). One of the largest obstacles faced in designing an APSS is that of its analysis. Introducing the adaptive component to the controller compromises somewhat one's ability to apply some of the well-established analysis tools of the power industry, with eigenvalue analysis being chief among them. Although there is an abundance of work that has succeeded principally in designing an adaptive PSS by employing a variety of modern control and identification schemes, there is often a curious absence of a robust, analytical investigation of their performance. [6, 9, 10, 17] Rigorous analysis is essential to understanding what takes place in the inner workings of such a controller. This understanding is, in turn, a prerequisite if an APSS is ever to be considered a serious alternative to the conventional PSS. Bearing this in mind, the author has made a concerted effort to conduct quantitative analysis whenever possible.

The remainder of Chapter 1 describes the chosen identification and control algorithms as well as reasons for their selection. In Chapter 2, investigation of the chosen identification scheme is conducted. The focus will be on the application of the algorithm to a power system network. The intended use of the identified model for the design of an APSS is considered throughout. Chapter 3 examines the control algorithm that is employed. Again, the focus is placed on issues pertaining to the application of that algorithm to a power system control problem. An analytically derived model of the process is used throughout. Chapter 4 takes the first step towards the realization of an adaptive PSS by combining the results of Chapters 2 and 3. Controllers are synthesized using identified process models instead of analytically calculated ones. Finally, Chapter 5 and Chapter 6 give some conclusions and a discussion of future work that could build off the foundation established here.

1.2 Background

1.2.1 Algorithm Selection

Designing an adaptive controller requires the selection of both an identification scheme and a control scheme. An exhaustive search for the best choices was not conducted as part of the research. Instead, the chosen algorithms were selected based on their reported advantages and virtues. A linear identification algorithm and a linear control algorithm were desired in order to facilitate analysis using traditional methods such as eigenvalue analysis.

Identification Scheme

The chosen identification scheme was the Recursive Least-Squares (RLS) algorithm. Its choice was largely due to its simplicity and its extensive use in other adaptive control problems.

Control Scheme

The control scheme selected for implementation uses the Generalized Predictive Control (GPC) algorithm. While many modern control theory algorithms have been developed over the years most lack robustness in regards to dealing with either minimum-phase plants, plants of unknown order or time varying plants. The GPC algorithm is reported to be better equipped to deal with the combination of these issues than most other algorithms [3, 4]. Additionally, a GPC controller is synthesized based on knowledge of the IO relationship of the underlying process. The adopted RLS identification algorithm, like most other identification algorithms, yields only an IO relation of the process being identified and provides no direct information about its inner workings. This coincides well with the formulation the GPC algorithm, which requires only an IO model of a process to synthesize a controller for it. In contrast, the traditional PSS design approach requires information about the phase lag between a machine's electric torque, T_e, and its exciter's voltage reference signal, V_{ref}. While it would be presumptuous to state that this information could not be extracted from the IO model, doing so would certainly be more complicated than simply applying the IO model directly.

1.2.2 Recursive Least-Squares (RLS) Algorithm

Below, the development of the RLS algorithm is summarized. This algorithm models an SISO relationship of a given process by fitting it as best as it can to a difference equation of the form given in equation 1-1. In this equation *y(t)* is a process output and *u(t)* is a process input. Before identification can take place, assumptions regarding model order must be made. This is done by selecting values for n_a and n_b of equation 1-1 such that $n_a \geq n_b$. In general it can be shown that modeling an m^th^ order process using a difference equation of the form given in equation 1-1 is accomplished by setting $n_a = m$ and $n_b = m-1$.

$$A(z^{-1})y(t) = B(z^{-1})u(t-1) \qquad \textbf{[1-1]}$$

where

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a}$$
$$B(z^{-1}) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b}$$

The development of the RLS algorithm follows from *least-squares estimation*, a non-recursive version of the former. Accordingly, a description of least-squares estimation is a logical starting point for explaining the RLS algorithm.

Least-Squares Estimation

The first step is to cast equation 1-1, above, into a form which emphasizes the information that is known as well as the information which is being searched for. Equation 1-2 below, accomplishes this. Here, *θ(t)* is a vector containing the difference equation coefficients targeted for identification and *x(t)* is a vector of past inputs and outputs of the process, which presumably would be measured.

$$y(t) = \mathbf{x(t)}^{\mathbf{T}}\boldsymbol{\theta}\mathbf{(t)} + e(t) \qquad \textbf{[1-2]}$$

where

$$\boldsymbol{\theta}\mathbf{(t)}^{\mathbf{T}} = [-a_1, \ldots, -a_{n_a}, b_0, \ldots, b_{n_b}]$$
$$\mathbf{x(t)}^{\mathbf{T}} = [y(t-1), \ldots, y(t-n_a), u(t-1), \ldots, u(t-n_b-1)]$$

Equations 1-1 and 1-2 are identical except for the additional *e(t)* term found in the latter which has been added to account for modeling error. If written for subsequent time steps, equation 1-2 can be used to construct the matrix equation of 1-3. Here, N is the total number of samples taken and must be greater than or equal to $n_a + n_b + 1$, the number of coefficients being identified. If this requirement is not met then the problem is improperly constrained and an infinite number of solutions exist.

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \mathbf{x^T(1)} \\ \mathbf{x^T(2)} \\ \vdots \\ \mathbf{x^T(N)} \end{bmatrix} \hat{\boldsymbol{\theta}}\mathbf{(t)} + \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(N) \end{bmatrix} \Rightarrow \mathbf{Y} = \mathbf{X} \cdot \hat{\boldsymbol{\theta}}\mathbf{(t)} + \mathbf{E} \qquad \textbf{[1-3]}$$

The principle of least-squares estimation is to select $\hat{\boldsymbol{\theta}}\mathbf{(t)}$ such that the sum of all the squared error terms is minimized. This objective can be expressed as a mathematical formulation using the cost function given in equation 1-4.

$$J = \sum_{i=1}^{N} e(i)^2 \qquad \textbf{[1-4]}$$

The goal then becomes to minimize the value of *J*. It can be shown that this optimization can be achieved using the simple manipulations of equation 1-5.

$$\hat{\boldsymbol{\theta}} = [\mathbf{X^T X}]^{-1}[\mathbf{X^T Y}] \qquad \textbf{[1-5]}$$

Recursive Least-Squares Algorithm

The burden associated with the calculation of equation 1-5 becomes prohibitive for a large number of samples, N. In order to be practically useful any identification technique must not entail an excessive computational burden. For this reason iterative schemes are ideally suited for adaptive controller applications. Fortunately, the least-squares estimation method described above can cast into a recursive form. Derivation details can be found in [2] and a summary is given below.

At time t+1:

(i) Construct **x(t+1)** using past values of the sampled signals *y(t)* and *u(t)*

(ii) Estimate the current modeling error, *e(t+1)* using

$$e(t+1) = y(t+1) - \mathbf{x}^{\mathbf{T}}\mathbf{(t+1)}\hat{\boldsymbol{\theta}}\mathbf{(t)} \qquad [\,1\text{-}6\,]$$

(iii) Form **P(t+1)**, the covariance matrix, using

$$\mathbf{P(t+1)} = \mathbf{P(t)}\left[\mathbf{I} - \frac{\mathbf{x(t+1)x^T(t+1)P(t)}}{1+\mathbf{x^T(t+1)P(t)x(t+1)}}\right] \qquad [\,1\text{-}7\,]$$

The covariance matrix is the $[\mathbf{X}^{\mathbf{T}}\mathbf{X}]^{-1}$ factor of equation 1-5 calculated at time *t*. Selection of its initial value is discussed in [2] and is dependent on the uncertainty in the initial parameter estimates.

(iv) Update $\hat{\boldsymbol{\theta}}\mathbf{(t)}$, the parameter estimates using

$$\hat{\boldsymbol{\theta}}\mathbf{(t+1)} = \hat{\boldsymbol{\theta}}\mathbf{(t)} + \mathbf{P(t+1)x(t+1)}e(t+1) \qquad [\,1\text{-}8\,]$$

(v) Wait for the next time step then repeat steps (i) to (v).

Estimation Adaptability

In order to improve the adaptability of the estimated parameters the objective function of the RLS algorithm is modified slightly. The parameter estimates, $\hat{\boldsymbol{\theta}}\mathbf{(t)}$, are now selected to minimize a weighted sum of the squared errors in the predictions they yield. The largest weighting is placed on the most recent error. This objective can be expressed as the new cost function of equation 1-9.

$$J = \sum_{i=1}^{N} \lambda^{N-i} e(i)^2 \qquad [\,1\text{-}9\,]$$

A similar derivation as before leads to a recursive algorithm identical to the previous one except the covariance matrix, *P(t+1)*, is calculated in a slightly different manner. The modified covariance matrix is given in equation 1-10.

$$\mathbf{P(t+1)} = \frac{\mathbf{P(t)}}{\lambda}\left[\mathbf{I} - \frac{\mathbf{x(t+1)x^T(t+1)P(t)}}{\lambda+\mathbf{x^T(t+1)P(t)x(t+1)}}\right] \qquad [\,1\text{-}10\,]$$

The symbol λ is known as the *forgetting factor* and generally, the smaller its value the faster the parameter estimates can adapt to changes. Introduction of the forgetting factor does, however, entail the added drawback of making the parameters more susceptible to any noise in the signals used to construct the model.

System Excitation

In order to effectively identify a process there must be sufficient variability in the IO data set being fed to the RLS algorithm. Failure to satisfy this condition will lead to poor identification and may even cause algorithmic instability. [2] Often a supplementary signal is applied to the process in order to ensure that it remains sufficiently excited. This signal is commonly referred to as a *dither signal*. Care must be exercised in selecting the magnitude of the dither signal because it should be large enough to help with identification yet small enough so that it does not adversely affect the normal operation of the process. Also, the spectrum of the dither signal must be such that it excites the process modes of interest.

Challenges

Despite its widespread use, the RLS algorithm does have several limitations. One of these is that it becomes unstable when identifying an unstable plant. This instability is a numerical problem. Numerical problems can also arise when using a forgetting factor less than one. Additionally, the RLS algorithm requires the pre-selection or assumption of a model order for the process being identified. Often, however, accurate knowledge of the process' complexity is unavailable.

1.2.3 Generalized Predictive Control (GPC) Algorithm

The GPC algorithm is a modified form of the generalized minimum variance algorithm. That algorithm uses a linear model of the process intended for control to select a control increment which acts to minimize the difference between the process' next predicted output and its desired output. The general form of this model is given in equation 1-11.

$$\begin{aligned} & A(z^{-1})y(t) = B(z^{-1})u(t-1) \\ & \textit{where} \\ & \quad A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a} \\ & \quad B(z^{-1}) = b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b} \end{aligned} \qquad \textbf{[1-11]}$$

Generalized predictive control expands on the idea of minimum variance control by carrying out the optimization over a finite number of future predictions. [3,4] This objective can be expressed mathematically as the cost function of equation 1-12. In it *y(t+j)* is a prediction of the process output at the *(t+j)*$^{\text{th}}$ sampling, *w(t+j)* is the set-point value of the desired output at the *(t+j)*$^{\text{th}}$ sampling and *Δu(t+j)* is the calculated control increment intended to make the actual output approach the desired set-point value.

$$J_1 = E\left\{ \sum_{j=N_1}^{N_2} \left(y(t+j) - w(t+j) \right)^2 + \sum_{j=1}^{N_u} \alpha(j) \cdot \Delta u(t+j)^2 \right\} \qquad \textbf{[1-12]}$$

N$_1$ is the *minimum costing horizon* and is usually given a value of either one or, when known, the dead-time of the plant. N$_2$ is the *maximum costing horizon* and is often set as the rise time of the plant. The symbol *α(j)* represents *control penalty* and takes on values greater than or equal to zero. The control penalty limits the activity of the controller and is usually set as constant. Finally, N$_u$, is referred to as the *control horizon*. It denotes the length of the temporal window during which the controller is assumed to be active. All future control increments beyond the control horizon are assumed to be zero. Choice of control horizon should be restricted such that its value must be less than the maximum costing horizon.

The work which follows will make use of a simplified version of equation 1-12 where the minimum costing horizon is assumed to be one and the maximum costing horizon will be renamed the *output horizon*, denoted simply by N. Moving forward, both the output set-point and the control penalty will be taken as constants. These simplifications result in the cost function of equation 1-13.

$$J_1 = E\left\{\sum_{j=1}^{N}\left(y(t+j)-w\right)^2 + \sum_{j=1}^{N_u}\alpha\cdot\Delta u(t+j)^2\right\} \qquad \textbf{[1-13]}$$

where

α	is the control penalty	N_u	is the control horizon
N	is the output horizon	w	is the output set-point

The first step in the optimization of the cost function of equation 1-13 requires the construction of three sets of polynomials. They are constructed by making use of the process model in conjunction with the control parameters. Each polynomial is a function of the backward shift variable z^{-1} and their general forms are given below in equations 1-14 and 1-15.

For j = 1…N

$$G_j(z^{-1}) = g_{j0} + g_{j1}z^{-1} + \cdots + g_{j(n_b+j-1)}z^{-(n_b+j-1)} \qquad \textbf{[1-14]}$$

$$E_j(z^{-1}) = e_{j0} + e_{j1}z^{-1} + \cdots + e_{j(j-1)}z^{-(j-1)} \qquad \textbf{[1-15]}$$

$$F_j(z^{-1}) = f_{j0} + f_{j1}z^{-1} + \cdots + f_{j(n_a)}z^{-n_a} \qquad \textbf{[1-16]}$$

The E polynomials serve as intermediaries and are calculated only to derive the G polynomials. The coefficients of E_j and F_j polynomials are calculated in a recursive manner. Defining F_1 and E_1 according to equations 1-17 and 1-18 starts the recursions.

$$F_1(z^{-1}) = f_{10} + f_{11}z^{-1} + \ldots + f_{1n_a}z^{-n_a} = z\left(1-\tilde{A}\right) \qquad \textbf{[1-17]}$$

$$E_1(z^{-1}) = e_0 = 1 \qquad \textbf{[1-18]}$$

$$\textit{where} \quad \tilde{A} = \tilde{a}_0 + \tilde{a}_1 z^{-1} + \ldots + \tilde{a}_{n_a+1} z^{-(n_a+1)} = \left(1-z^{-1}\right)A\left(z^{-1}\right)$$

The coefficients for the remaining E_j and F_j polynomials can be found using equations 1-19 through 1-21.

$$e_{ji} = e_{(j-1)i} \qquad i = 0\ldots(j-2) \qquad \textbf{[1-19]}$$

$$e_{ji} = f_{(j-1)0} \qquad i = j-1 \qquad \textbf{[1-20]}$$

$$f_{ji} = f_{(j-1)(i+1)} - \tilde{a}_{i+1}\cdot f_{(j-1)0} \qquad i = 0\ldots n_a \qquad \textbf{[1-21]}$$

The G polynomials can subsequently be calculated using equation 1-22.

$$G_j(z^{-1}) = B(z^{-1}) \cdot E_j(z^{-1}) \qquad \textbf{[1-22]}$$

Having constructed the G and F polynomials, minimization of the cost function of equation 1-13 can proceed. The result of this optimization, which can be carried out using algebraic manipulations, is equation 1-23.

$$\hat{\mathbf{u}} = \left[\mathbf{G_1}^{\mathbf{T}}\mathbf{G}_1 + \lambda\mathbf{I}\right]^{-1}\mathbf{G_1}^{\mathbf{T}}(w - \mathbf{f}) \qquad \textbf{[1-23]}$$

Here, **û**, is an N x 1 vector of projected control increments. The current control increment, *Δu(t)*, is taken as the first element of this vector. Its values are integrated in order to arrive at the final control signal. **G₁** is a N x N$_u$ matrix constructed from the elements of the G polynomials. Similarly, **f** is a vector of polynomials which are functions of the backward shift variable z^{-1} and which are constructed using the elements of both the G and F polynomials. The construction of both **G₁** and **f** is given below in equations 1-24 and 1-25 respectively.

$$\mathbf{G_1} = \begin{bmatrix} g_{10} & 0 & \cdots & 0 \\ g_{21} & g_{20} & \cdots & 0 \\ \vdots & \cdots & \cdots & . \\ g_{n(n-1)} & \cdots & & g_{n0} \\ \vdots & \cdots & & \vdots \\ g_{N(N-1)} & g_{N(N-2)} & \cdots & g_{N0} \end{bmatrix} \qquad \textbf{[1-24]}$$

$$\mathbf{f} = \begin{bmatrix} f(t+1) \\ f(t+2) \\ \vdots \\ f(t+N) \end{bmatrix} = \begin{bmatrix} \left[\ g_{11}z^{-1} + \cdots + g_{1(n_b+1)}z^{-n_b}\ \right]\Delta u(t) + \left[\ f_{10} + \cdots + f_{1(n_a)}z^{-n_a}\ \right]y(t) \\ \left[\ g_{22}z^{-1} + \cdots + g_{2(n_b+2)}z^{-n_b}\ \right]\Delta u(t) + \left[\ f_{20} + \cdots + f_{2(n_a)}z^{-n_a}\ \right]y(t) \\ \vdots \\ \left[g_{NN}z^{-1} + \cdots + g_{N(N+n_b)}z^{-n_b}\right]\Delta u(t) + \left[f_{N0} + \cdots + f_{N(n_a)}z^{-n_a}\right]y(t) \end{bmatrix} \qquad \textbf{[1-25]}$$

Analysis

It was previously stated that an emphasis on analytical assessment was going to be a focal point of this research. In order to apply some traditional stability assessment techniques, an equivalent transfer function for a designed GPC controller is required. This is done fairly simply by first rewriting equation 1-23 above as follows:

$$\hat{\mathbf{u}} = \mathbf{K} \cdot (w - \mathbf{f}) \qquad \textbf{[1-26]}$$

$$where \quad \mathbf{K} = \left[\mathbf{G_1}^{\mathbf{T}} \mathbf{G_1} + \lambda \mathbf{I} \right]^{-1} \mathbf{G_1}^{\mathbf{T}}$$

Here, **K** is a N_u x N matrix. The current control increment, *Δu(t)*, being the first element of û, can be expressed as the vector equation given in equation 1-27 or equivalently by equation 1-28.

$$\Delta u(t) = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1N} \end{bmatrix} \cdot \left[w - \begin{bmatrix} f(t+1) & f(t+2) & \cdots & f(t+N) \end{bmatrix}^T \right] \qquad \textbf{[1-27]}$$

$$\Delta u(t) = \left(\sum_{i=1}^{N} K_{1i} \right) w - \sum_{i=1}^{N} K_{1i} \cdot f(t+i) \qquad \textbf{[1-28]}$$

When the set-point, *w*, is set to zero and the polynomials of *f* are substituted out using equation 1-25, a transfer function of the form given in equation 1-29 can be constructed.

$$\frac{\Delta u(t)}{y(t)} = \frac{n_0^{'} + n_1^{'} z^{-1} + \cdots + n_{n_a}^{'} z^{-n_a}}{m_0^{'} + m_1^{'} z^{-1} + \cdots + m_{n_b}^{'} z^{-n_b}} = \frac{N^{'}}{M^{'}} \qquad \textbf{[1-29]}$$

where

$$m_0^{'} = 1 \qquad \textbf{[1-30]}$$

$$m_j^{'} = \sum_{i=1}^{N} K_{1i} g_{i(j-1+i)} \qquad j = 1 \ldots n_b \qquad \textbf{[1-31]}$$

$$n_j^{'} = \sum_{i=1}^{N} K_{1i} f_{ij} \qquad j = 0 \ldots n_a \qquad \textbf{[1-32]}$$

Appending an integrator to the transfer function of equation 1-29 yields the complete transfer function for the designed GPC controller, *u(t)/y(t)*. Here *y(t)* is the process output/controller input and *u(t)* is the controller output. Having constructed this transfer function it then becomes a simple matter to compute the closed-loop poles or to incorporate it into a system state equation. (see appendix B)

1.2.4 Continuous versus Discrete Control

Beyond the design differences that will be discussed in the coming chapters, a conventional PSS and the proposed GPC PSS differ fundamentally in that the former is designed in the continuous-time domain while the latter is designed in the sample domain. It is important for the reader to understand the implications of such a difference as it pertains to both the design and analysis of the controllers.

Design

The overwhelming majority of design and analysis work done in the power industry makes use of continuous-time models. One of the first tasks in designing a GPC PSS therefore was the acquisition and/or construction of discrete-time models. In this work, either identified models or analytically calculated models were used during the synthesis of the GPC controller. Arriving a discrete model in each of these cases required different steps. Cases where the model was identified were simple because a discrete model was output directly from the RLS algorithm. Analytically calculated discrete-time models, on the other hand, were obtained through discretization of analytically calculated continuous-time models. This task was achieved using equation B-1 of appendix B.

The situation often arose where common components, usually modeled as continuous-time domain transfer functions, needed to be included into the constructed discrete models. In such cases the *bilinear transformation* was useful. [5] The transformation is given by equation B-2 of appendix B where its use in the discretization of a first order washout filter is demonstrated.

A designer must be aware of certain pitfalls that arise as a result of using discrete models. When a controller is designed in the continuous-time domain, there is no risk of losing sight of the fact that the underlying process is also continuous. When the model of the process is discretized it is implicitly assumed that the process can be controlled by changing the input at a rate no faster than the sampling frequency. Consider the case of an unstable continuous-time process that demands a certain rate of response in order to stabilize it. If the process is not modeled using a fast enough sampling time it will become impossible for any controller designed using that model, regardless of how the control parameters are selected, to stabilize the process. An exceptional case would exist when a dual rate system was employed. With such a system the controller output can change at a rate different than the sampling rate of the model used in the synthesis of that controller [15, 17]. Clark's original GPC algorithm, however, does not allow for this and thus special care must be taken when choosing the sampling rate.

Analysis

Since the proposed controller being designed was a discrete one, the analyzes presented in this work are done exclusively in the sample domain. This required some adjustments to the typical analysis approaches but nothing that could not be overcome. The eigen-properties of the system are largely unchanged by discretization. Although the eigenvalues of the discrete-time and continuous time state matrices were different, the damping, ξ, and the frequency of oscillation, ω, were the same. Calculation of these two metrics in the discrete-time domain, however, requires different equations than in the continuous-time domain. [8] See equations B-14 and B-15 of appendix B.

Comparing the performance of a continuous-time conventional PSS to that of a discrete-time GPC based PSS was a fairly simple matter. The traditional PSS was designed as per normal in the continuous-time domain and was incorporated into the system state equations. The modified state equations were subsequently discretized allowing for the comparison of the two closed-loop systems in the discrete-time domain. Figure 1-2 gives a synopsis of what was involved in the design, analysis and comparison of the traditional and GPC PSSs.

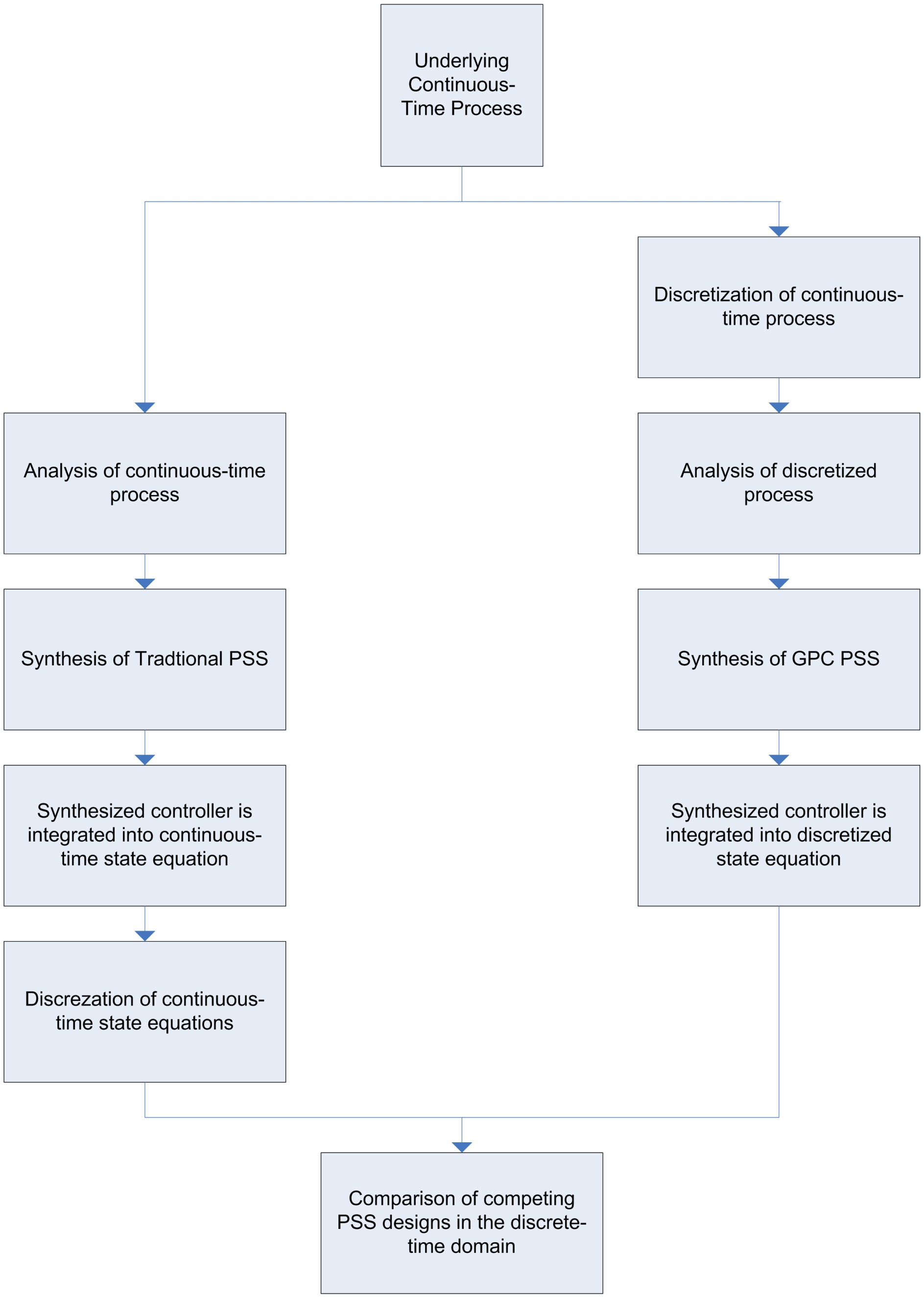
Underlying Continuous-Time Process
Discretization of continuous-time process
Analysis of continuous-time process
Analysis of discretized process
Synthesis of Tradtional PSS
Synthesis of GPC PSS
Synthesized controller is integrated into continuous-time state equation
Synthesized controller is integrated into discretized state equation
Discrezation of continuous-time state equations
Comparison of competing PSS designs in the discrete-time domain

Figure 1-2: Diagram showing the stages involved in the design, analysis and comparison of traditional and GPC PSSs.

1.2.5 Calculation/Simulation Platform

The analytical results of this research were obtained using code that was written in MATLAB®. All of the time-domain simulations were completed using Simulink® wherein the algorithms being researched were implemented using Simulink structures known as *S-functions*.

2 System Identification

The first step in the design of an adaptive power system stabilizer is the identification of the process to be controlled. Identification is done in the sample domain and the often used recursive least squares algorithm is the chosen scheme. The investigations of this chapter are based on the identification of a simple linearized one-machine infinite bus (OMIB) model. Many of the challenges associated with the application of the RLS algorithm are investigated. First simulation considerations such as sampling rate selection and measurement noise are identified and discussed. Subsequently the issues of signal quality, model adaptability and model order selection are examined in series.

2.1 Simulation Considerations

2.1.1 Identification Target

The SISO relation being identified throughout this chapter is $\Delta\omega/\Delta V_{ref.}$ Here, $\Delta\omega$ is the machine speed deviation and ΔV_{ref} is the deviation in the field reference voltage. The process by which $\Delta\omega/\Delta V_{ref}$ was chosen as the IO relation around which the controller is designed is detailed in the next chapter.

2.1.2 Sampling Rate

In light of the interest in low-frequency oscillations, which typically range from 0.5 to 3Hz, Nyquist's theorem is satisfied by conservatively choosing a sampling frequency of 20Hz ($T_S = 0.05$s). Although sufficient for identification purposes, 20Hz proved insufficient for control purposes. Once an investigation of the control algorithm began, it was quickly discovered that being able to change the control output 20 times per second was inadequate. This rate did not allow for the consistent stabilization of many of the constructed test systems. The controller needed to be capable of acting faster in order to effectively control those processes. The sampling rate was therefore increased from 20Hz and eventually selected as 100Hz ($T_S = 0.01$s). This rate is used throughout the report.

2.1.3 Dither Signal

A dither signals was employed in order to ensure persistent excitation of the simple network used in this work. Assessing the impact of this supplementary excitation signal is a complicated task. Certain types of dither signals are more appropriate for identification than others and the optimum frequency for a given signal is something that must be researched. The investigations that follow will address neither the selection of the best type of dither signal, nor the selection of the best frequency. Instead, we focus on assessing the impact of the dither signal amplitude.

A pseudo-random binary sequence (PRBS) was used as the dither signal because it is widely accepted as an excellent signal for this purpose due to its rich spectral content. [2] The PRBS is a deterministic signal that can be generated in many ways. The particular implementation chosen for the work presented here is described in [7]. The chosen sequence has a fundamental period of approximately 31 seconds and during that interval the possibility of switching occurs every second. These choices seem suitable given that the interest is in low-frequency power system phenomena. The switching that occurs will excite the modes of interests and the intervals between the switches are long enough to allow for a few cycles of oscillation to be observed. Figure 2-1 shows a sample of a PRBS. For the sake of consistency the same PRBS is used in all the simulations of this chapter.

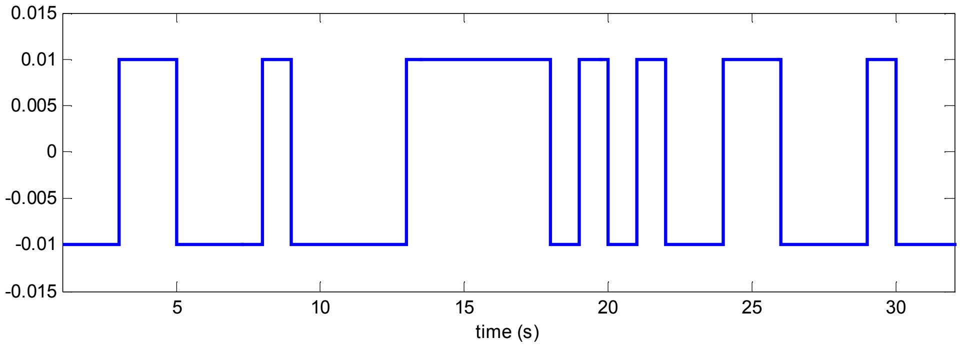

Figure 2-1: An example of a pseudo-random binary sequence (PRBS).

A practical dilemma presented by the use of the dither signal concerns whether or not it can be continuously applied to the process being identified. In section 2.1.1 ΔV_{ref} was established as the input of the process model being constructed and accordingly the dither signal will be applied to it. Deliberately and continuously superimposing a small disturbance on the voltage reference signal of a generator's exciter may not be acceptable to industry. Evaluating whether or not continuous identification is acceptable is beyond the scope of this work. The ramifications of either decision, though, should be considered. If continuous identification is judged acceptable the time constraints on speed of identification are significantly less stringent than if identification is allowed only during short intermittent periods. The existence of a time constraint will dictate selection of many of the identification parameters. Particularly that of the forgetting factor since it has a significant effect on the speed of identification. Despite the reality of this dilemma, it does not affect the general observations which can be made about the RLS algorithm and thus will only be considered further as a discussion point.

2.1.4 RLS Startup

An important consideration in the identification process is the selection of an initial covariance matrix, **P(0)**. An improper selection will lead to a very slow start-up. The initial magnitudes of the elements of the covariance matrix are a direct reflection of the amount of uncertainty that exists in the initial parameter estimates. If the initial estimates are nearly correct then small initial values for the covariance matrix elements are appropriate. Likewise, incorrect initial estimates require large initial values for the covariance matrix elements. Figure 2-2 compares the time evolutions of the RLS estimated parameters that result when the covariance matrix is assigned small and large values of 10^2 and 10^{14} times the identity matrix respectively. In this figure the initial choice of covariance matrix influences the initial speed of parameter convergence in the expected fashion. The only issue to be settled is that of determining a good initial value. The method employed here was to assign the RLS algorithm a small initial covariance matrix and have it run from a dead start until the parameter estimates began to settle. The initial covariance matrix is then assigned a value comparable to that which developed as the parameter estimates converged.

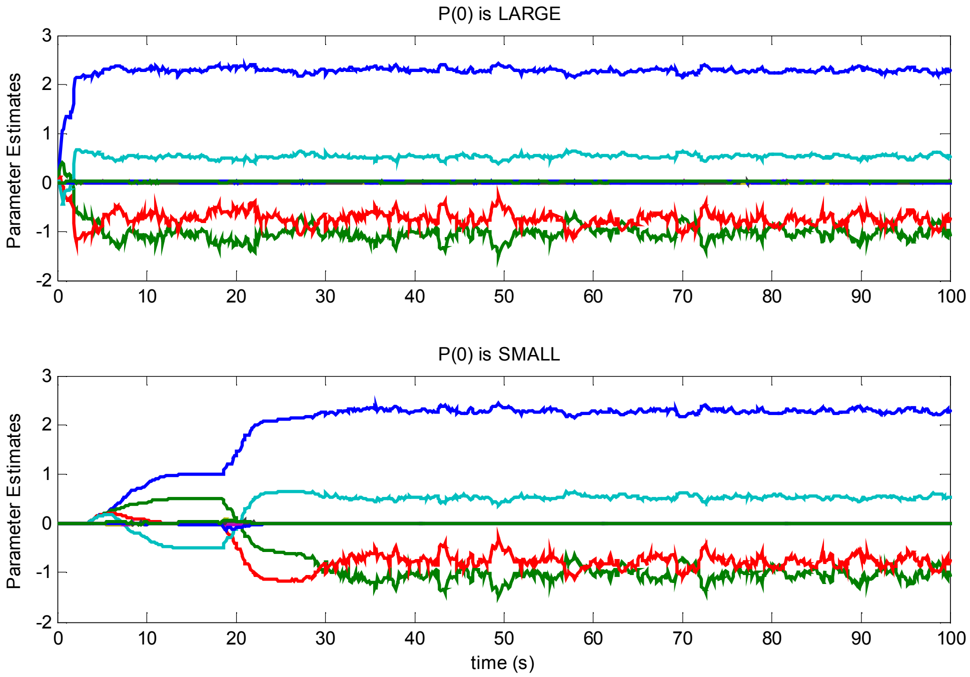

Figure 2-2: Demonstration of the effect that small and large initial covariance matrices have on the initial rate of parameter convergence.

2.1.5 Parameter Settling and Buffering

Early in the course of this research it was realized that whenever the signals used for process identification were noisy or distorted the estimated parameters yielded by the RLS algorithm would not settle. Even small amounts of noise and/or distortion triggered this effect. Clearly, this presents an obstacle in regards to application of the estimated parameters for controller design purposes. Using a time-varying system model to design a controller would result in a time varying controller. This is not attractive from the standpoint of system analysis. The use of traditional analysis tools, such as eigenvalue analysis, would be limited since they are applicable only to Linear Time-Invariant (LTI) systems. Using this foresight, an attempt to circumvent the problem of a time varying controller was made by periodically buffering the parameter estimates of the RLS algorithm. In this way, the controller could be updated at periodic intervals. In between updates, the controller remains static and traditional stability analysis would be valid. This approach will be referred to as the *constant forgetting factor buffering scheme*. Figure 2-3 shows the time evolution of the RLS parameters when such a strategy is implemented. Capture of the estimated parameters occurs on the positive edge of the rectangle waveform shown. The interval between bufferings is chosen arbitrarily as 40s and the forgetting factor is set to 0.985.

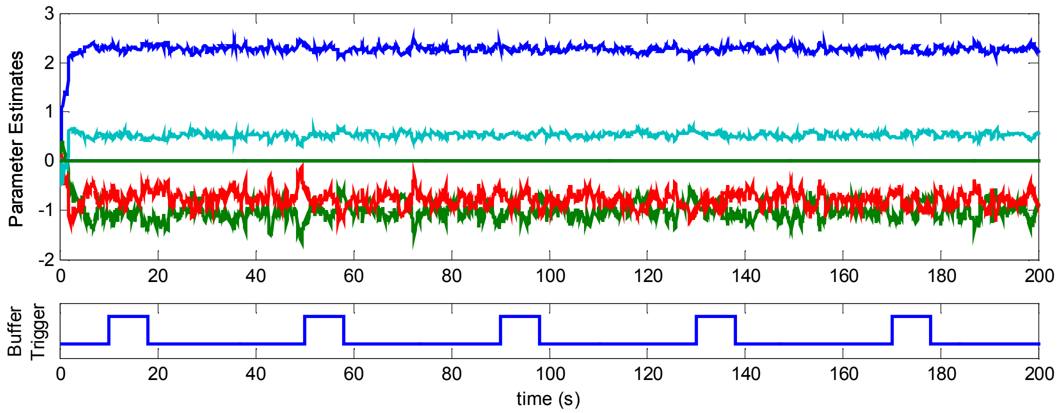

Figure 2-3: Time evolution of the RLS estimated parameters using the constant forgetting factor buffering scheme.

Once buffered, the parameters can then be used for controller synthesis. The only problem with this approach is that with a forgetting factor less than unity, noise in the signals used to construct the model tend to have a significant impact on the least-square solution. Consequently, the parameter values that are buffered are values which may be significantly affected by noise. A solution is found by toggling the forgetting factor between two different values. Assigning the forgetting factor a value less than one for a period of time before switching it to one allows the parameter estimates to settle. Before the forgetting factor is switched back to its non-unity value, the RLS algorithm's estimated parameters can be sampled and buffered. In this fashion the adaptability of the estimated model is maintained while the impact of noise on the parameter estimates used for either controller synthesis or for analysis purposes is kept low. The described approach will henceforth be referred to as the *variable forgetting factor buffering scheme*. The interval during which the forgetting factor takes on a value less than one will be referred to as the *adaptive window*. Usually the adaptive window will account for only a minor portion of each buffering cycle. The diagram of figure 2-4 indicates how the forgetting factor is toggled between two different values and figure 2-5 shows the time evolution of the estimated parameters when a variable forgetting factor buffering scheme is used.

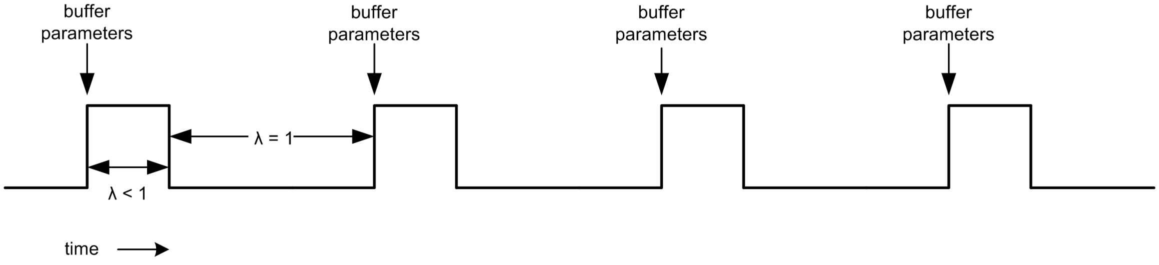

Figure 2-4: Trigger Pattern for Parameter Freezing

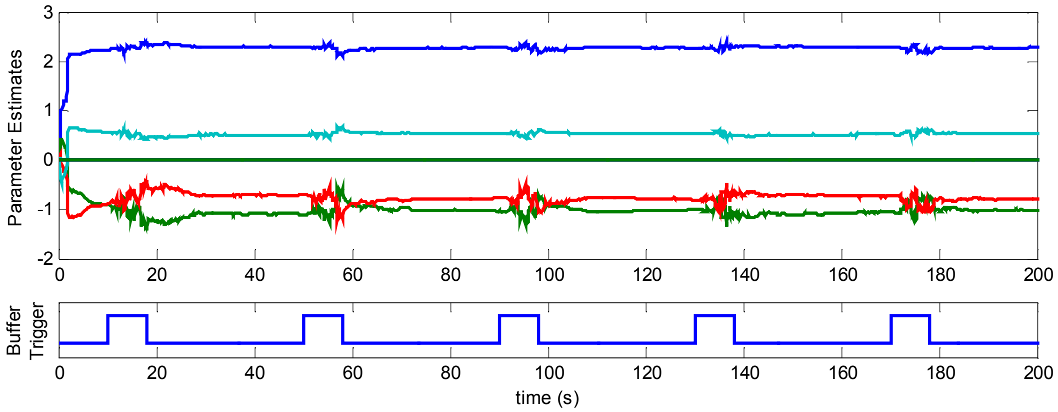

Figure 2-5: RLS Estimated Parameters using the variable forgetting factor scheme.

2.1.6 General Investigative Approach

In the process of conducting this research, choices regarding the conditions under which identification would take place had to be made. Originally, it was thought that presenting an analysis of the RLS algorithm would be best done by examining its performance under a series of different conditions. The first might be a noiseless environment, the second might include noise/distortion of the signals fed to the identification algorithm and the third might be the same as the second except the noise corrupted signals would be filtered before being passed to the identification algorithm. The diagrams of figure 2-6 illustrate the proposed experimental setups.

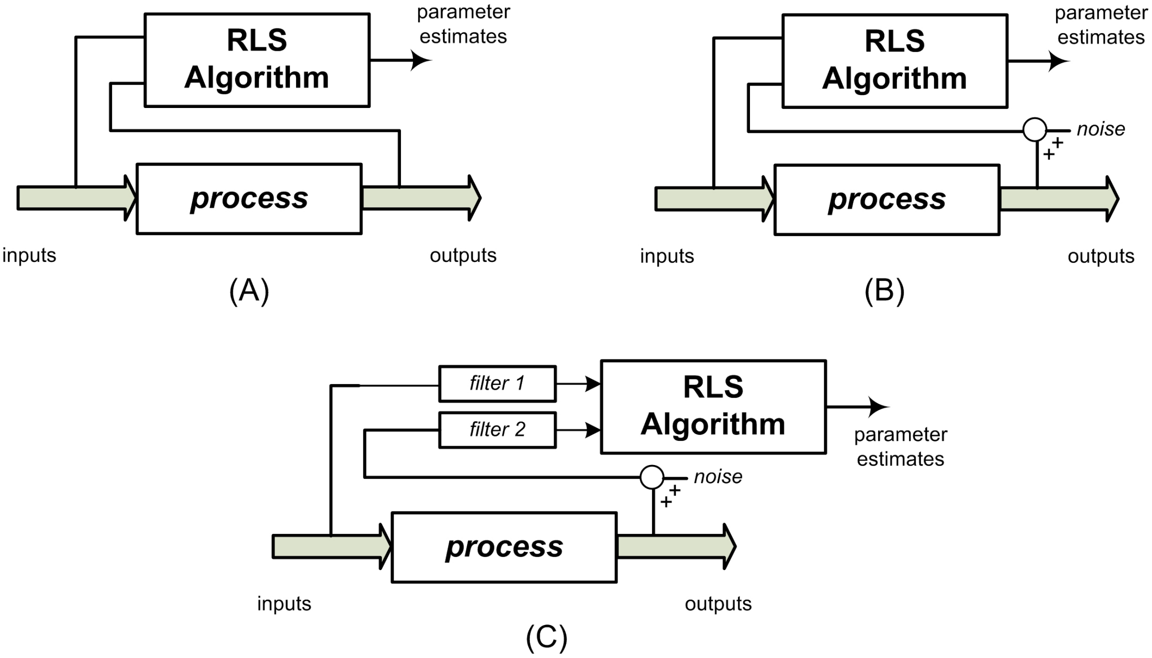

Figure 2-6: Block diagram representing different candidate setups for investigation of the RLS algorithm.

Thoughts of such an approach, however, were eventually abandoned and instead a conscious decision was made to focus on the setup of diagram C where the signals were corrupted and pre-filtering was employed. A study of the situation where noise corrupted signals would be applied to the RLS algorithm without pre-filtering was seen as frivolous. Even in preliminary investigations the benefits gained from employing filters could not be disregarded. The filters facilitated, without sacrificing identifiability, use of supplementary excitation signals hundreds of times smaller than would otherwise have been possible. Such an advantage is especially relevant for the particular identification problem presented here. The smaller the supplemental excitation required, the smaller the disturbance to the nominal operating condition of the network and the more palatable the proposed scheme becomes. As far as identification in an ideal noiseless environment is concerned, it is not used in this chapter to a great extent because in practical terms such an environment does not exist. All real measurements contain some noise and/or distortion. Still a brief discussion of how the RLS algorithm performs in a noise-free environment is instructive if only because it provides a baseline for comparison against identification in a noisy environment. A summary of the results when identification is carried out in an ideal environment is given below.

Noiseless Environment

When identification was carried out in a noiseless environment and the model order was correctly assumed the magnitude of the excitation dither signal did not appear to tangibly affect the accuracy of the identified parameters. The estimated parameters tended to converge with zero ripple to their expected, analytically calculated, values and accordingly so did the dominant process poles. The only relation of note was that an increase in the size of the dither signal led to faster convergence of the estimated parameters. When the process was either under-modeled or over-modeled the estimated parameters converged with negligible ripple to values for which there was no analytical basis for comparison. Still the dominant complex pole pair of these estimated models remained fairly well identified regardless of the model order assumption. The single exception was when the process was modeled as a 1st order system. The observed failure under this assumption, however, was not unexpected because despite having the physics of the process dictate the dominant poles as complex, the assumed model order is such that the algorithm can only identify them as real.

2.1.7 Experimental/Simulation Setup

Having decided to focus on evaluating the RLS algorithm when it is subjected to filtered, noise corrupted inputs; there remain a few further issues to be discussed.

Filter Specification.

The simulations conducted in this work included only noise/distortion at the output of the process. Although noise might appear at the input as well it is not of great concern because the system's long time constants prevent such noise from affecting the output to any great extent and therefore it is reasonable to neglect it. Bearing this in mind the filter is designed to remove high frequency noise at the output of the process without causing undue distortion of the output signal. The filter used was of the low-pass, Finite Impulse Response (FIR) variety. Its order was selected to be 99, a fairly high number so that the drop-off was sharp. The sampling frequency was selected to be 200Hz, which is higher than the highest frequency component in the simulations and thus avoids aliasing problems. The filter cutoff frequency was selected as 4Hz. Practically, it is very uncommon to use such a high order filter but the results should hold for most other types of filters which provide good performance.

If diagram C of figure 2-6 is revisited it becomes clear that the task of assigning *filter 1* remains. An intuitive choice for *filter 1* would be a pure delay that would compensate for the time lag introduced by the FIR filter. Delaying the process input signal appropriately ensures that it is properly synchronized with the process output signal before being passed to the RLS algorithm. Such synchronization is a requirement for accurate identification. It turns out, however, that selecting *filter 1* the same as *filter 2* is better suited to minimizing the effect that the filters have on the RLS parameter estimates. If the model order assumption is correct, there exists no noise, and the bandwidth of the dither signal isn't too severely restricted by the filter; then making *filter 1* and *filter 2* identical would lead to ideal identification. This statement follows from the block diagrams of figure 2-7. Diagram A of figure 2-7 would lead to ideal identification and diagrams B and C follow from it if the process and filters are assumed to be linear. Given this finding, it is evident that the same filter should be applied to both the process input and output signals.

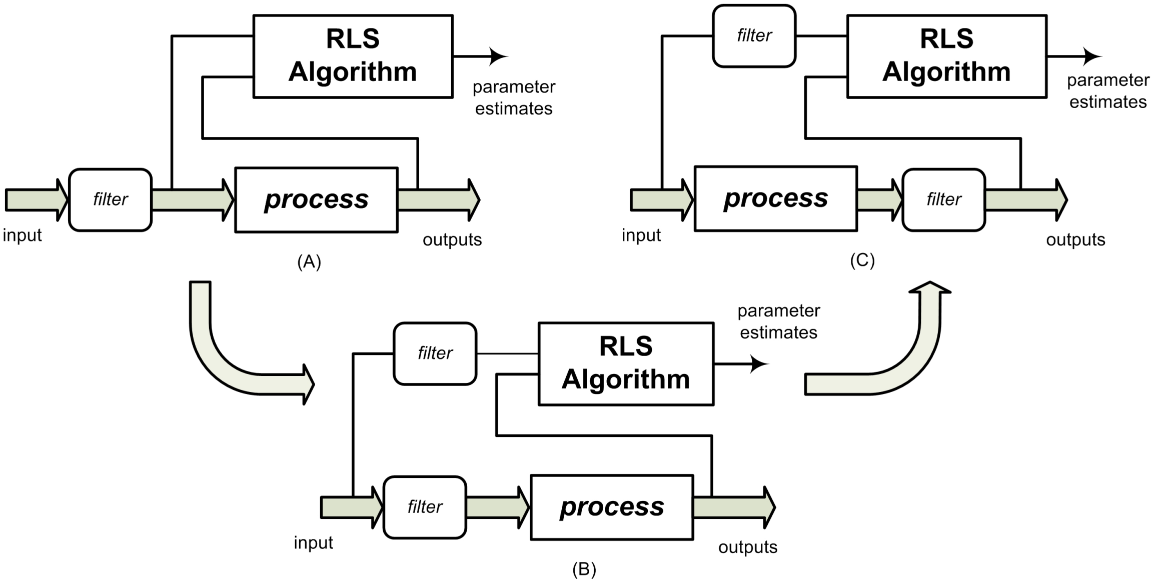

Figure 2-7: Block diagrams showing how selecting *filter 1* and *filter 2* the same is the best means of minimizing their impact on the estimated parameters.

It should be stressed to the reader that the filters themselves add distortion to the signals they filter. Generally, their use is a benefit but in the extreme it hampers identification in cases where the pre-filter Signal-to-Noise Ratio (SNR) is large. In such a situation the SNR decreases after filtering. Still, practically speaking, this negative side-effect should not become an issue because the signals will never be so clean that passing them through a good filter degrades them to the point where parameter identification would be notably compromised.

Noise Specifications

Quality of identification is significantly affected by the type, level and frequency of the noise; among other things. Conducting a robust investigation of each of these factors is a daunting task. A simplified approach is therefore adopted. The noise source is assumed to be such that, for the most part, the perturbation it introduces can be separated from the actual signals of interest. The results presented in the next few sections would prove valid for different noise sources if this assumption holds. The goal of a separable noise source was realized by using a high frequency, normally distributed (Gaussian) random signal. It had a zero mean value and a variance of 1e-11. The noise source was kept fixed throughout this chapter. It was applied only at output of the process based on the justification given above. Figure 2-8 below is provided to give the reader feel for the signals used during the simulations. The signal in question is the synchronous generator machine speed deviation. Column A compares the noise-corrupted signal to the clean signal. Column B compares the filtered noise-corrupted signal to the clean signal. The filter used is of the FIR variety and is described above. The SNR calculated is based on pre-filter signals.

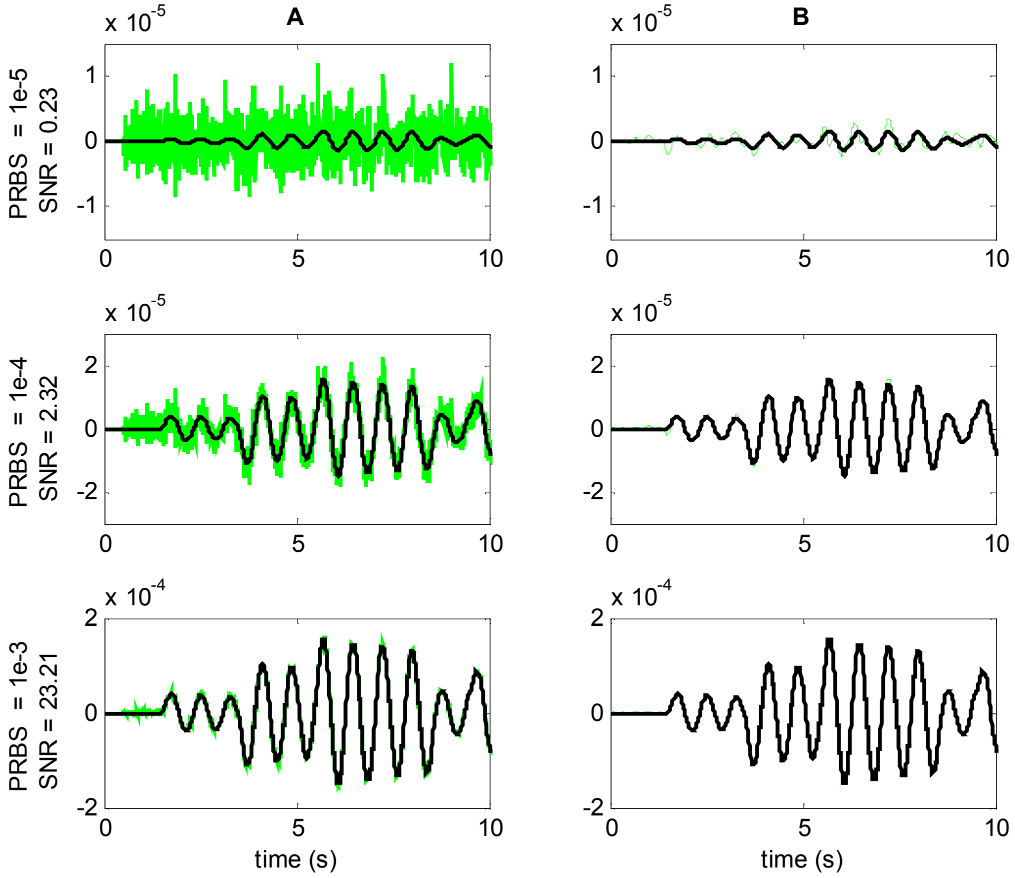

Figure 2-8: Column A shows a comparison of the noisy and clean machine speed deviation signals resulting from application of PRBSs of varying amplitude. Column B shows how the filtered signals compare to the clean signals. The variance of the output noise is kept constant at 1e-11.

2.2 Signal Quality

The quality of the signals used by the RLS algorithm is a determining factor in the quality of the identification achieved. Those signals containing little or no noise/distortion would be classified as high quality and their use leads to good identification. Contrarily, those containing substantial amounts of noise/distortion are of low quality and are less suitable for identification purposes. Because the process being identified is deliberately excited using a dither signal, the size of the output signal can be manipulated. Assuming that the noise source is predominantly independent of these manipulations, signal quality can therefore be controlled.

The most general approach is to evaluate the quality of identification as a function of the SNR of the signals used to construct a given model. This is especially true in light of limited foreknowledge about the levels of noise/distortion that can be expected in a practical situation. The noise source described in section 2.1.7 was kept fixed and the dither signal amplitude was increased; thereby simulating an increase in the SNR. Since all of the simulations make use of LTI models, care to ensure that the selected excitation levels were reasonable was not required to obtain accurate results. Calculation of the SNR was done discretely from time-domain simulations ran for 200s using equations B-16 and B-17 of appendix B.

Particular attention should be paid to how well the estimated dominant closed-loop mode corresponds with the actual dominant closed-loop mode. The dominant mode is selected because it is strongly associated with the dynamic behavior of the process and because of its predictability vis-à-vis that of the estimated parameters, which do not converge to their analytically calculated values for all but the most simple of cases.

2.2.1 Simulation Results

Table 2-1 lists the results of a series of simulations that were carried out. The model order is correctly assumed to be four for all the simulations. The variable forgetting factor buffering scheme was employed with the forgetting factor assuming a value of 0.990 during the adaptive widow. Test System 1 of appendix A is the process being identified and the parameter estimates are assigned initial values of zero. The calculated SNRs both prior to and after filtering are given. The error in the real and imaginary components of the estimated dominant closed-loop mode is also shown. The actual dominant mode of the process is 0.99595 ± 0.081791i. The errors from three separate buffered parameters are provided to emphasize that the errors shown are the result of noise corruption and not because the parameters have yet to converge.

PRBS	**Pre-filter**	**Post-filter**	***Estimated Error (%)***						**Estimated Dominant**
amplitude	**SNR**	**SNR**	**REAL**			**IMAG**			**Mode**
			3	***4***	***5***	***3***	***4***	***5***	***5***
1.00E-05	0.06	0.82	4.52	4.40	4.40	72.18	68.47	78.03	0.95209 + 0.14561i
1.19E-05	0.09	1.17	4.36	4.23	4.31	64.94	61.22	71.51	0.95305 + 0.14028i
1.43E-05	0.12	1.67	4.10	3.95	4.13	56.28	52.66	63.42	0.95484 + 0.13367i
1.70E-05	0.18	2.38	3.73	3.55	3.84	46.66	43.34	53.99	0.95769 + 0.12595i
2.03E-05	0.25	3.39	3.24	3.04	3.44	36.91	34.13	43.85	0.96171 + 0.11766i
2.42E-05	0.36	4.83	2.69	2.48	2.93	28.06	25.95	34.00	0.96675 + 0.109600i
2.89E-05	0.51	6.88	2.13	1.94	2.38	20.75	19.27	25.41	0.97226 + 0.102570i
3.46E-05	0.72	9.80	1.62	1.47	1.85	15.10	14.09	18.57	0.97755 + 0.096976i
4.12E-05	1.03	13.96	1.21	1.09	1.39	10.87	10.19	13.39	0.98211 + 0.092743i
4.92E-05	1.47	19.88	0.89	0.79	1.02	7.76	7.30	9.58	0.98576 + 0.089625i
5.88E-05	2.10	28.30	0.64	0.56	0.74	5.49	5.19	6.80	0.98857 + 0.087357i
7.02E-05	2.99	40.25	0.46	0.40	0.53	3.86	3.66	4.80	0.99066 + 0.085720i
8.38E-05	4.26	57.21	0.33	0.28	0.38	2.69	2.56	3.37	0.99218 + 0.084547i
1.00E-04	6.07	81.23	0.23	0.20	0.27	1.86	1.77	2.35	0.99329 + 0.083711i
1.19E-04	8.65	115.17	0.16	0.14	0.19	1.27	1.22	1.62	0.99407 + 0.083118i
1.43E-04	12.32	162.93	0.12	0.09	0.13	0.86	0.82	1.11	0.99464 + 0.082700i
1.70E-04	17.56	229.82	0.08	0.06	0.09	0.57	0.55	0.75	0.99503 + 0.082405i
2.03E-04	25.02	322.84	0.06	0.04	0.06	0.37	0.35	0.50	0.99531 + 0.082199i
2.42E-04	35.66	450.93	0.04	0.03	0.04	0.24	0.22	0.32	0.99551 + 0.082055i
2.89E-04	50.82	624.94	0.03	0.02	0.03	0.15	0.13	0.20	0.99564 + 0.081956i
3.46E-04	72.42	857.07	0.02	0.01	0.02	0.09	0.07	0.12	0.99574 + 0.081888i
4.12E-04	103.20	1159.30	0.01	0.01	0.01	0.05	0.04	0.06	0.99581 + 0.081843i
4.92E-04	147.08	1540.70	0.01	0.00	0.01	0.03	0.02	0.03	0.99585 + 0.081814i
5.88E-04	209.60	2003.30	0.01	0.00	0.01	0.02	0.01	0.01	0.99588 + 0.081796i
7.02E-04	298.70	2538.20	0.00	0.00	0.00	0.02	0.00	0.01	0.99591 + 0.081787i
8.38E-04	425.68	3124.00	0.00	0.00	0.00	0.02	0.00	0.01	0.99592 + 0.081782i
1.00E-03	606.64	3727.90	0.00	0.00	0.00	0.02	0.01	0.01	0.99593 + 0.081781i
1.19E-03	864.52	4313.60	0.00	0.00	0.00	0.02	0.01	0.01	0.99594 + 0.081782i
1.43E-03	1232.00	4848.50	0.00	0.00	0.00	0.02	0.01	0.01	0.99594 + 0.081784i
1.70E-03	1755.80	5311.10	0.00	0.00	0.00	0.02	0.01	0.01	0.99594 + 0.081786i

continued…

continued from…

2.03E-03	2502.20	5692.70	0.00	0.00	0.00	0.02	0.01	0.00	0.99594 + 0.081788i
2.42E-03	3565.80	5995.40	0.00	0.00	0.00	0.02	0.01	0.00	0.99595 + 0.081789i
2.89E-03	5081.70	6228.20	0.00	0.00	0.00	0.02	0.01	0.00	0.99595 + 0.081791i
3.46E-03	7241.90	6403.00	0.00	0.00	0.00	0.02	0.01	0.00	0.99595 + 0.081793i
4.12E-03	10320.00	6532.00	0.00	0.00	0.00	0.02	0.01	0.00	0.99595 + 0.081794i
4.92E-03	14708.00	6625.90	0.00	0.00	0.00	0.02	0.01	0.00	0.99595 + 0.081795i
5.88E-03	20960.00	6693.70	0.00	0.00	0.00	0.01	0.01	0.01	0.99595 + 0.081796i
7.02E-03	29870.00	6742.30	0.00	0.00	0.00	0.01	0.01	0.01	0.99595 + 0.081796i
8.38E-03	42568.00	6777.00	0.00	0.00	0.00	0.01	0.01	0.01	0.99595 + 0.081797i
1.00E-02	60664.00	6801.70	0.00	0.00	0.00	0.01	0.01	0.01	0.99595 + 0.081797i

Table 2-1: Accuracy of estimated dominant closed-loop mode as a function of the quality of the signals used to construct the estimated model. Indicated is the dither signal level, the pre- and post-filter SNRs, the error in real and imaginary components for three different sets of buffered estimated parameters and the estimated dominant mode from the last set of buffered parameters.

For the moment there is little comment to be made about these results. The one notable feature is that the RLS algorithm clearly has more difficulty accurately identifying the imaginary component of the dominant mode than it does identifying its real component. Further discussion of the implications of this observation is left for section 2.2.2. The reader should also make note of the benefits offered by pre-filtering the inputs to the RLS algorithm. For signals with a low SNR the use of the filter dramatically increased it. Of course, when signals had a low noise content use of the filter appeared to degrade them, decreasing their SNRs. Still, this latter negative implication only occurs for impractically clean signals and the degraded post-filtering signals are nonetheless of excellent quality.

In order to attach some qualitative meaning to the results of table 2-1, a comparison of the time response from selected estimated models is given in figure 2-9. The response of the analytically calculated model is also included and acts as a baseline for comparison. All models are subject to the same disturbance, a 1.0 pu step increase to ΔV_{ref} applied at time t = 1s.

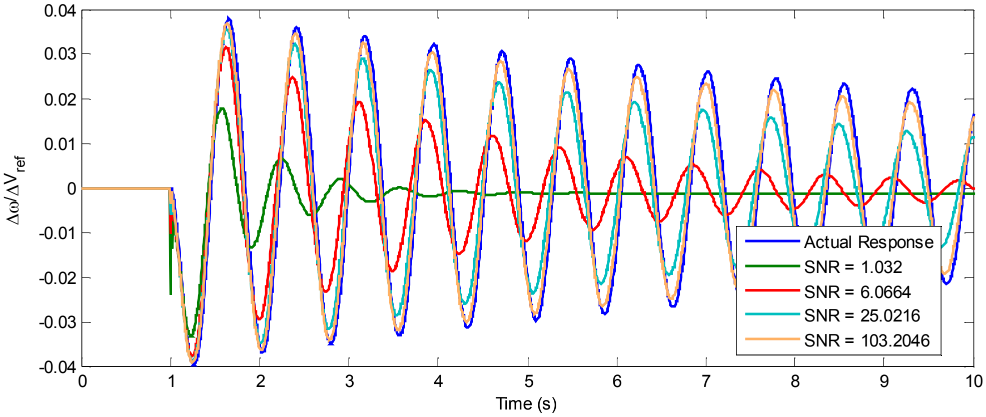

Figure 2-9: Comparison of the time responses from selected RLS estimated models.

Expectedly, the ability of the estimated models to accurately predict the process output is a strong function of the SNR of the signals used to construct them. Besides some initial high frequency spiking, for a short period of time following the disturbance the responses of the estimated models were faithful to those of the actual model. The duration over which any given estimated model maintained this accuracy grew as the SNR of the signal used in their construction was increased. This trend continued until the response from the estimated and actual models became indistinguishable. Still, despite having a fairly high pre-filter SNR of ~100, the estimated model output differs visibly, though not substantially, from the correct response.

As a final comment, the reader should be aware that signal quality impacted the speed with which parameters estimates converged to accurate values. Like before, this accuracy is measured in terms of the degree to which the estimated dominant closed-loop mode would converge to its expected value. Generally, the higher the signal quality, the faster the identification.

2.2.2 Discussion and Analysis

Despite having constructed and compared a series of estimated models the answer to the question *'What level of signal quality is required to achieve adequate* *identification?'* remains unclear. It is not a simple question to answer because it is not a straightforward matter to define what *'adequate identification'* is.

The most simplistic definition would be *'Any identification which yields a model whose output is faithful to that of the actual process'*. Defining *'adequate identification'* in this manner is not encouraging. Results for section 2.2.1, specifically those of figure 2-9, show that impractically large SNRs are needed to construct a model capable of faithfully reproducing the output of the actual process. Achieving a pre-filter SNR of 100, for instance, would likely require a level of supplemental excitation that would significantly disrupt the function of the AVR. This problem is all the more discouraging when considering that this high level of signal quality was needed despite the simplicity of the system being identified.

Fortunately, there exists a better definition of *'adequate identification'* which considers the broader control problem. When defined as *'Any identification which allows for the construction of a model suitable for the synthesis of an effective controller'* it opens the door to the possibility of using estimated models which at first glance may be dismissed as inadequate. In the previous subsection reference was made to the short-lived ability of some of the poorer estimated models to produce an output faithful to the actual output. This property may be exploitable when designing a controller using these estimated models; especially in lieu of the predictive control algorithm that will be deployed. In any case, drawing conclusions regarding the signal quality needed to synthesize an effective controller would be premature at this point and is reserved for Chapter 4.

For the moment, *'adequate identification'* will be defined as *'Any identification which yields models that can reasonably differentiate between different operating points'*. Ultimately, satisfying this particular definition is required if the application of an adaptive controller is to be justifiable. There is no use for an adaptive controller that cannot accurately track operating point changes. A static controller would suffice.

Figure 2-10 below plots out the error in damping ratios and oscillation frequencies from the estimated dominant open-loop modes of table 2-1. Although the plots of figure 2-10 will change somewhat if a different process is being identified, the amount of variation is not significant. Damping ratio and oscillation frequency were the two metrics chosen because they are widely held to be defining characteristics of a given power network. Using the plots of figure 2-10 the reader can approximate the level of signal quality needed to identify the indicated system properties within a pre-specified error tolerance.

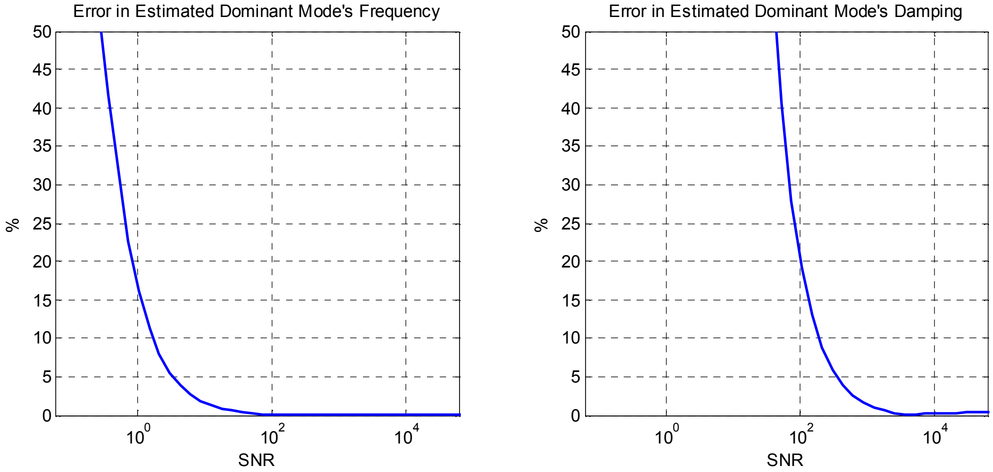

Figure 2-10: Plots of the error in the estimated dominant mode's frequency and damping as a function of the SNR of the signal input to the RLS algorithm.

The most obvious trend from these plots is that accurate identification of the dominant mode's damping requires much cleaner signals than are needed for accurate identification of the mode's oscillation frequency. The reader should be somewhat alarmed by this finding. For the identification problem presented here, ensuring that the estimation error in the dominant mode's damping falls below 5% requires a pre-filter SNR greater that 300. In contrast, a 5% estimation error in the dominant modes frequency can be achieved using a pre-filter SNR of ~3.

Based on a requirement to identify both oscillation frequency and damping, these results suggest that the RLS algorithm is incapable of differentiating between different operating points without the use of impractically high-quality signals. This casts doubts as to the viability of the algorithm’s use in an adaptive PSS. On the other hand, consider for a moment that the basic design objective of a PSS is to introduce an electric torque which is in phase with machine speed deviations. If the accurate knowledge of the machine’s oscillation frequency is not known then it will be impossible to generate an electric torque in phase with it. Contrarily, inaccurate knowledge of the machine’s damping neither hampers nor enhances the ability to generate the desired torque. Based on this reasoning accurate identification of the system’s oscillation frequency is much more critical than identification of its damping in regards to the design of an effective PSS. The results above are then subject to reinterpretation. It can then be concluded that the RLS algorithm *‘can reasonably differentiate between different operating points'* without the use of impractically high quality signals. Here, the standard of reason necessitates differentiation of those operating point properties that are most critical to the design of an effective controller.

Additional Comments:

This section has dealt with the excitation signal purely from the system identification perspective. No regard has been given to the practical constraints attached to its application. Most notable is the constraint that restricts the magnitude of the exciting signal so that it does not disturb, too greatly, the normal functioning of the process. In the context of the adaptive PSS problem, the supplementary excitation signal must be small enough so that the voltage regulation function of the AVR is not compromised.

Also, based on the research conducted, SNR alone appears to be an insufficient metric to predict the quality of identification. The nature of the distortion is also important. Although not shown in this report, the SNR needed to achieve a certain quality of identification without a filter was orders of magnitudes larger than the SNR needed to achieve a similar quality of identification with one. The observed sensitivity to distortion type might be eliminated if a more statistically robust method of calculating SNR were to be used.

2.3 Model Adaptability

In order to make the RLS estimated parameters adaptable, the original cost function was modified to place more emphasis on accurately fitting the most recently sampled data. The result of this modification was the introduction of the parameter, λ, known as a forgetting factor which must be assigned a value less than or equal to unity. Regardless of the particulars of the identification problem at hand, certain observations hold true. If the forgetting factor is too close to one then the adaptability of the estimated model is relatively slow. At the other end of the spectrum, a forgetting factor that is too small will increase adaptability but can lead to instability of the identification algorithm. This instability is a numerical problem. Ultimately, the selection of a forgetting factor becomes one of compromise between adaptability and stability.

Before proceeding with an investigation of the two conflicting topics mentioned above, it is important to recognized that all choices of forgetting factor are highly dependent on the choice of sampling frequency. The general rule is that the faster the sampling frequency, the faster the identification of a given process can be achieved. This trend continues until a plateau is reached where sampling at a higher rate does not yield additional information about the process. Also, the higher the sampling frequency the larger the minimum forgetting factor needed to maintain algorithmic stability.

2.3.1 Adaptability

In order for the reader to appreciate the impact that a forgetting factor can have on the adaptability of the estimated models, figures 2-11 and 2-12 are provided. In each, the ability of the identification algorithm to track changes in the real and imaginary parts of the dominant process mode is shown. Like before, particular attention is given to the dominant mode because of its strong correlation with system performance. The forgetting factors used in plots of figures 2-11 and 2-12 are 0.990 and 1 respectively. The process being identified is Test System 1 of appendix A. At time t = 70s, the system operating point changes when the power output of the machine decreases from 0.6 pu to

0.5 pu. The pre-filter SNR was set impractically large in the simulations shown but the speed of adaptation did not change significantly for lower signal qualities. The parameter estimates are initially set to zero and the constant forgetting factor buffering scheme is used. The first few estimates of the dominant mode have been omitted because their error content obscures interpretation of the plots.

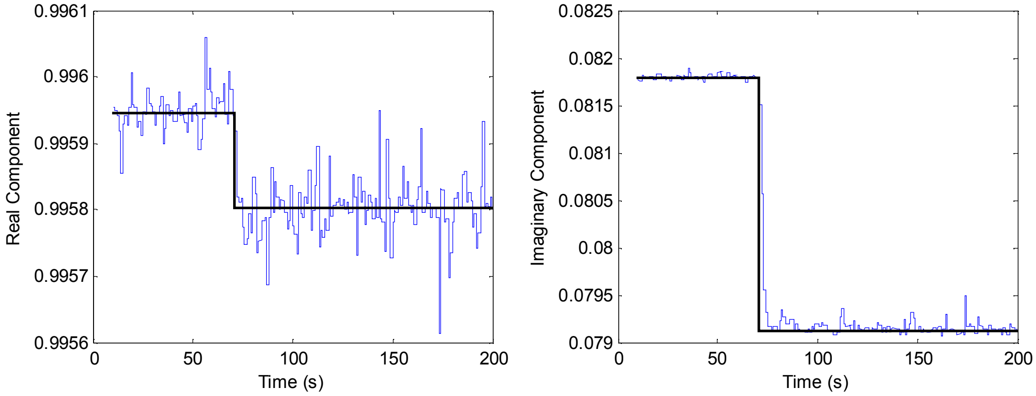

Figure 2-11: RLS algorithm employing a *constant forgetting factor buffering scheme* and a forgetting factor of 0.990 tracks changes in the real and imaginary parts of the process' dominant pole.

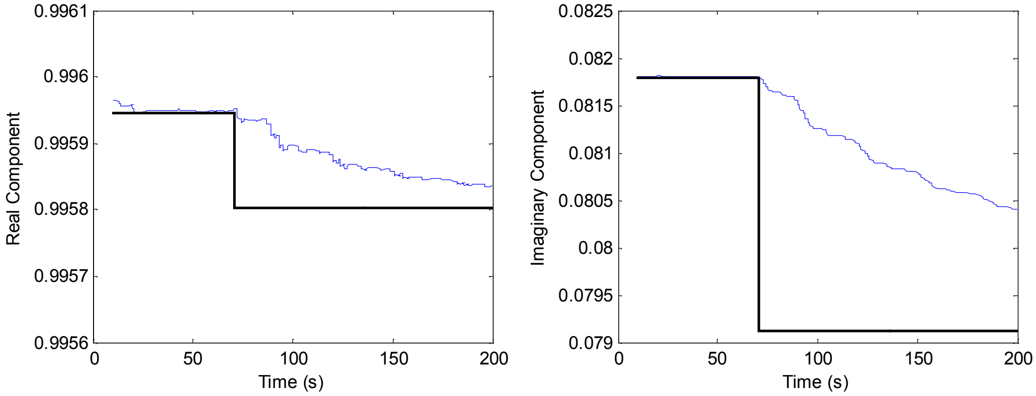

Figure 2-12: RLS algorithm employing a *constant forgetting factor buffering scheme* and a forgetting factor of 1.000 tracks changes in the real and imaginary parts of the process' dominant pole.

In comparing the two figures it becomes clear that use of a non-unity forgetting factor can significantly increase the speed of adaptation. Of course the exact rate of convergence will also depend on a number of other factors including, most notably, the selected sampling rate. Still, figure 2-11 demonstrates that adaptation can occur quite quickly using a non-unity forgetting factor. This finding validates, to an extent, the variable forgetting factor buffering scheme proposed to limit the impact of noise on the models used for the synthesis of controllers. If the adaptation takes place very quickly then the only thing sacrificed by toggling between forgetting factors is the ability to adapt immediately to a change in the process. This, however, is not a serious concern since the adaptive PSS is only intended to adapt to slow changes in operating point. Figure 2-13 shows how the real and imaginary part of the estimated dominant mode converge to their true values when the variable forgetting factor buffering scheme is used. The results fall somewhere in between those of figure 2-11 and figure 2-12 in terms of adaptability and variability. When the change in the process occurs at time t = 70s, there is an initial lag in the adaptability of the estimated model until an adaptive window is entered at time t = 90s. From that point forward, identification of the dominant model remains fairly good.

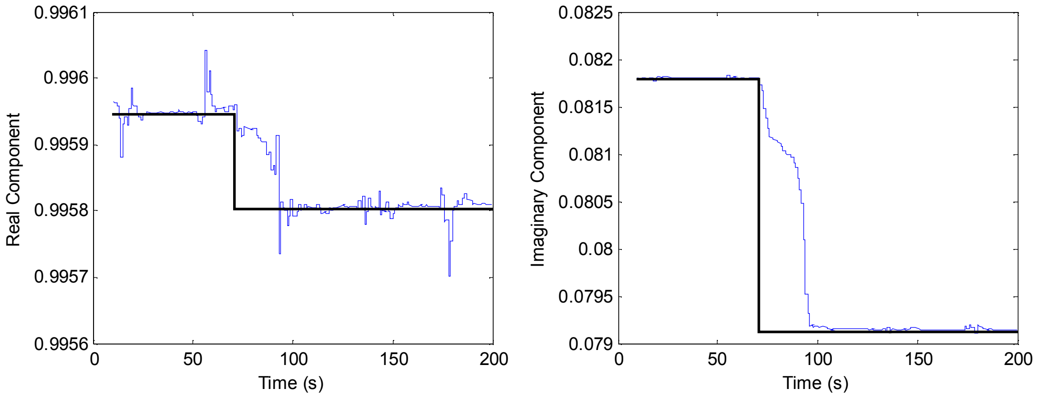

Figure 2-13: RLS algorithm employing a *variable forgetting factor buffering scheme* and a forgetting factor of 0.990 tracks changes in the real and imaginary parts of the process' dominant pole

2.3.2 Stability

The relationship which exists between stability and forgetting factor is a fairly simple one. The smaller the forgetting factor, the less stable the RLS algorithm. A measure of this instability is the trace of the algorithm's covariance matrix. As the forgetting factor is decreased, the elements of the covariance matrix increase in size as signaled by the growth in its trace. As the size of the covariance matrix increases, so does the probability that the RLS algorithm will become unstable as a result of numerical overflow. Table 2-2 shows how the forgetting factor impacts the average value of the covariance matrix trace. A variable forgetting factor buffering scheme was employed in the simulations used to derive the results. The supplementary excitation level was given a large value of 0.01 pu. As expected decreasing the forgetting factor triggered an increase in the average value of the covariance matrix trace which eventually led to instability. With the variable forgetting factor scheme, the trace cycles through periods of growth and decay. The periods of growth occur when the forgetting factor takes on a value less < 1 and those of decay occur when the forgetting factor is set to unity. As a result of the cycling, the average value of the trace is reduced over when the constant forgetting factor buffering scheme is used.

λ	average trace value	
1.000	1.1187E+13	*stable*
0.995	4.6555E+13	*stable*
0.990	8.6193E+13	*stable*
0.985	1.2750E+14	*stable*
0.980	1.6990E+14	*stable*
0.975	2.1366E+14	*stable*
0.970	2.5980E+14	*stable*
0.965	3.6921E+14	*unstable*
0.960	3.7449E+14	*unstable*
0.955	9.1672E+14	*unstable*
0.950	4.5548E+15	*unstable*
0.945	3.2228E+16	*unstable*
0.940	2.5287E+17	*unstable*
0.935	2.0532E+18	*unstable*
0.930	1.6392E+19	*unstable*

Table 2-2: Average value of the covariance matrix trace as a function of forgetting factor. The stability of the algorithm for each choice of forgetting factor is indicated.

2.3.3 Discussion

The conflicting desires of having an adaptable yet stable identification algorithm were examined in this section. If the results of table 2-2 are reconciled with those from section 2.3.1, an interesting observation can be made. The plots of figure 2-11 show that the RLS algorithm was able to adapt very quickly to a change in the process when the forgetting factor was assigned a value 0.990. As remarked above, the speed of adaptation did not change significantly as the excitation level was adjusted. From this, it can be surmised that use of a forgetting factor near 0.990 is generally adequate to achieve fast adaptation. Table 2-2 shows that the RLS algorithm became unstable for some forgetting factor value between 0.965 and 0.970. These forgetting factors are a safe distance from the value of 0.990 which led to good adaptability of the algorithm. Generalizing these results, it can therefore be stated that good adaptability is achievable without incurring too great a risk of algorithmic instability.

Ultimately, choice of an appropriate forgetting factor is dependent on other design choices. These choices include, among others, the selection of sampling period, length of the adaptive window, and whether or not continuous identification is employed. What can be stated, though, is that selection of its value does not require any great precision. So long as the time constraints on the speed of identification are not excessively stringent, there will be many suitable values for the forgetting factor that provide good adaptability and that do not compromise the stability of the algorithm.

2.4 Model Order Selection

Mention was made in the background section of the need to select or assume the order of the model prior to identification. Power systems present a particular challenge in that they are typically very large, complicated and generally of unknown order. As a result, any model order assumption is unlikely to be representative of the actual complexity of the process being identified. An assumed model order that allows for the construction of a model that can capture the dominant dynamics of the underlying process must be settled for.

In the following section, an examination of the implications of model order assumption is carried out. Given this particular interest, the dither signal was selected large enough so that any failure to provide good identification would not be the result of insufficient signal quality. The process being identified is Test System 1 of appendix A. Each simulation lasts 200s and the results which follow are derived from the last available buffering of the estimated parameters. The initial parameter estimates are all set to zero. Like previous sections of this chapter, focus is given to the modes of the estimated model, and in particular the dominant modes, for two reasons. Firstly, because of their strong correlation to the dynamic behavior of the process and secondly because of their consistency relative to that of the estimated model coefficients themselves.

2.4.1 Simulation Results

The RLS algorithm was repeatedly applied to Test System 1, changing only the model order assumption with each application. The actual order of the linear state-space model being identified is four. Table 2-3 presents the calculated estimated open-loop poles as a function of the model order assumption. The analytically derived poles are 0.99595 ± 0.081791, 0.87523 and 0.69036.

Model Order Assumption	*Identified Eigenvalues*			
6	0.9959 ± 0.0818i	-0.4361 ± 0.3790i	0.8786	0.1592
5	0.9959 ± 0.0818i	-0.3608 ± 0.3118i	0.8800	
4	0.9959 ± 0.0818i	-0.5823	0.8812	
3	0.9959 ± 0.0818i	0.8831		
2	0.9968 ± 0.0815i			
1	0.9969			

Table 2-3: Estimated open-loop poles as function of model order assumption.

The performance of the RLS algorithm appears to be good. Generally, neither over-modeling or under-modeling of the process appears to significantly undermine the accuracy with which the dominant open-loop mode can be identified. It was only in the case of significant under-modeling, namely the assumption of a 1st order process, that the dominant mode was badly misestimated. Table 2-4 summarizes the error in the real and imaginary components of the estimated dominant mode for each model order assumption.

Model Order Assumption	Error in the dominant mode (%)	
	real	imaginary
6	0.0002%	0.0022%
5	0.0002%	0.0050%
4	0.0002%	0.0077%
3	0.0002%	0.0108%
2	0.0826%	0.3416%
1	0.0958%	100.0000%

Table 2-4: Error in the estimated dominant mode of the process.

Curiously enough the error in the estimated dominant pole decreases monotonically as the model order assumption is increased. The intuitive expectation would be that the error would be minimum for the correct model order assumption and would grow larger in the case of either increasing under-modeling or over-modeling. A probable explanation for what is observed is that the larger model order assumptions allow for some limited modeling of the distortion resulting from the application of the FIR pre-filters. Modeling this distortion then facilitates more accurate identification of the dominant mode.

In order to gain a better qualitative understanding of the impact of model order assumption on identification accuracy, the different models from tables 2-3 and 2-4 were subjected to the same disturbance and their responses were plotted. Results are given in figure 2-14. The disturbance takes the form of a 1 pu step increase in the model's input, ΔV_{ref}. Deviations of the estimated model responses from the analytically-calculated model's response only became significant as the process was increasingly under-modeled. The response of the 6th, 5th, 4th and 3rd order estimated models coincided quite well with the actual response of the process. From there the models became visibly less accurate. The response of the 2nd order estimated model, although overall quite different for the actual response of the process, compared favorably to it for a fraction of the first cycle. Finally, the response of the 1st order model was expectedly, based on the error in the identified dominant mode, quite inaccurate.

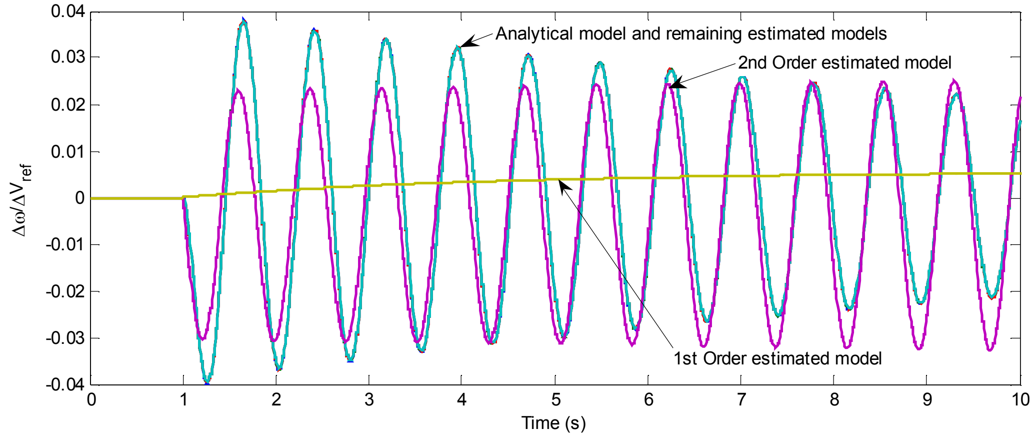

Figure 2-14: Comparison of the time responses from several estimated models and the actual model when each is subjected to an identical disturbance.

2.4.2 Discussion

Similar to the situation encountered when considering signal quality, it is still too early to define an adequate model assumption suitable for the synthesis of an effective controller. This is left for treatment in Chapter 4. What can be commented on, however, is the model order assumption needed to reasonably differentiate between operating points. Reiterating a comment from before, this differentiation is a requirement of any identification algorithm to be used in an adaptive scheme. Table 2-5 shows the error in the estimated dominant mode's damping and oscillation frequency as a function of the assumed model order. The errors reported in this table will change when a different process is being identified but not to any great extent. Therefore, generalizations can be made based on these results.

Model Order Assumption	Error in the dominant estimated mode's damping (%)	Error in the dominant estimated mode's frequency (%)
6	0.2967%	0.0020%
5	0.3645%	0.0048%
4	0.3800%	0.0075%
3	0.4605%	0.0105%
2	113.7900%	0.4255%
1	11590.0000%	96.2120%

Table 2-5: Error in the dominant estimated mode's damping and frequency of oscillation as a function of the model's assumed model order.

In most of the simulations, the errors precipitated by incorrect modeling order assumptions did not appear to tangibly affect the algorithm's ability to correctly identify the dominant mode. This began to change noticeably when the model order was assumed to be two, the number of dominant process eigenvalues. In this case, although the dominant modes frequency was well identified, a substantial error occurred in the estimation the dominant mode’s damping. Despite the significant error in the estimated damping of its dominant mode, the 2nd order estimated model still provides a reasonable prediction of the process' response to a step disturbance. This is evident from figure 2-14. Further reduction in the assumed model order led to the failure of the algorithm's capacity to produce a model capable of reasonably differentiating between operating points. This is evident by the sizable errors in both the estimated damping and estimated oscillation frequency of the dominant mode as well as by the time responses of the model presented in figure 2-14.

Using the same arguments advanced in section 2.2, the accurate identification of the process' oscillation frequency is judged more critical to the synthesis of an effective controller than identification of its damping. Assuming that this is true it is reasonable to conclude from the results above that while over-modeling is preferable to under-modeling, limited under-modeling is not a serious problem. This of course is a desirable conclusion since under-modeling is unavoidable in practice due to the size and complexity of real power networks.

A safe approach is to select the assumed model order slightly larger than the number of dominant eigenvalues. Selecting the model order exactly equal to the number of dominant eigenvalues will likely lead to less accurate estimations. This is because the RLS algorithm, trying to best model the IO relation it is charged with identifying, will distort the estimates of the dominant eigenvalues as it tries to fit the minor dynamics of the process. Setting the assumed model order larger than the number of dominant eigenvalues allows for limited modeling of the minor dynamics and thus leads to better modeling of dominant dynamics.

2.5 Summary

In this chapter, many aspects of the system identification problem were examined. It became clear that without a well defined approach a great number of simulations could be carried out without obtaining any meaningful results. The first order of business was to define the context of the investigation. This process is described in the paragraph below and involved making a number of assumptions.

First, the relation being sought was identified. The sampling rate used throughout this work was then specified along with the reasons for its selection. Certain characteristics of the supplementary excitation signal used to augment identification were described and issues pertaining to start-up of the algorithm were discussed. A scheme proposed to limit the impact of noise on estimated parameters intended for either analysis or for controller synthesis was devised. It was assigned the name *variable forgetting factor buffering scheme*. A decision was made to focus on investigating the performance of the RLS algorithm within a noisy, non-ideal identification environment in order to maximize the practical utility of results. The benefits of pre-filtering the inputs to the RLS algorithm were so great that investigations without the pre-filter were omitted from this report. Placement and configuration of the filters which minimized their impact on the parameter estimates was discussed and the specifications for the filters used throughout this report were given.

Having defined the context of our investigation, efforts were then turned to three issues of the identification problem deemed especially important. They are signal quality, model adaptability and model order selection. Each was allocated its own section in this chapter and was examined in detail. The performance of the algorithm can be studied in two ways. Of primary importance is accuracy. Study of signal quality and model order selection deal predominantly with accuracy of identification. Of secondary importance is the speed of identification which is dealt with for the most part in this work’s section on model adaptability. More than anything else the reader, having read this chapter, should gain a general understanding for the types of details which are relevant when considering signal quality, model adaptability and model order assumption. Some validation of the observations made in this chapter is reserved for Chapter 4 when estimated models are used to synthesize controllers.

Regarding accuracy of identification, at a minimum the RLS algorithm has to be able to differentiate between different operating points. Failure to do so makes its use in an adaptive controller unjustifiable. To this end the open-loop process’ frequency was seen as a property indicative of change in operating point and a property who’s accurate identification was required in order to synthesize an effective controller that can address that change. With this in mind signal quality became by far the most critical factor in determining the accuracy of the estimated model. Good identification was achievable using modest SNRs of ~5. Model order selection was not as large a determining factor in model accuracy. It was only when the assumed model order was set below the number of dominant open-loop poles that an unsuitable model would result. A safe approach is to set the model order assumption slightly higher than the number of dominant poles.

The speed of identification/adaptability was augmented through the use of a forgetting factor. The expected compromise between adaptability and stability arose and was studied. Choice of suitable forgetting factor was found to be dependent on a number of different factors; most notably sampling rate selection. Nevertheless, good adaptability could be achievable without incurring a significant risk of algorithmic instability. This finding should generally be true so long as the time restrictions placed on speed of identification are not too stringent.

3 Controller Synthesis

In Chapter 2 a first step towards designing an APSS was taken through investigation of the RLS identification algorithm. The GPC control algorithm is now examined in the same way. For the moment, gaining an understanding of the algorithm's performance when charged with the control task of a conventional PSS is the only interest. An analytically calculated model of the process to be controlled is thus used throughout the chapter. The combination of the results from this and the previous chapter is reserved for the next. Insofar as possible, the design development of the GPC PSS parallels that of a conventional stabilizer. What follows is a description of the major components of that development. Numerical details are omitted for the most part from the body of the work and are left for the appendices. Throughout this chapter the sampling period is selected as 0.01s, consistent with the choice made in Chapter 2. Simple investigations suggest that such a choice is sufficiently small to allow for good control.

This chapter begins with the basics of designing a GPC PSS given in section 3.1. While the basic design is fairly simple some adjustments were required to make the algorithm suitable for application as a PSS controller. Three significant obstacles arose, each of which is dedicated a section in this chapter. Section 3.2 addresses the problem of controller complexity. In section 3.3 tuning of the proposed controller is examined.

Section 3.4 deals with the last major design obstacle, the elimination of an offset in the controller signal. Section 3.5 contains a performance comparison between a traditional PSS and the designed GPC PSS. Finally, a summary given in section 3.6 concludes the chapter.

3.1 Basic Controller

The most succinct way to explain the process of synthesizing the basic GPC PSS is to do so by enumerating a series of steps:

1. The first step is to *identify the variable to be regulated as well as that used to effect the desired regulation*. These two variables are then respectively set as the input and output signals of the controller. Since the proposed GPC PSS would serve as a replacement for a traditional PSS and would be interfaced in a similar manner its output remains a modulating signal to an exciter's voltage reference signal, ΔV_{ref}. Although there is some flexibility in regards to the selection of the input signal the intuitive choice is to use the local machine's deviation in speed from nominal, $\Delta\omega$. The objective is to regulate $\Delta\omega$ to zero so it is reasonable to set it as the input of the controller.
2. Having defined what needs to be regulated and a means of achieving it, the next step is to *establish a discrete LTI approximation of the relation between these two signals*. While power systems are generally MIMO systems, the interest when designing a PSS is only in the two variables identified in step 1. As such, that particular component of the power system can be modeled as a SISO process.
3. Given that $\Delta\omega$ is being regulated, *the reference value should be set to zero* to ensure that the GPC algorithm works to minimize any deviations of machine speed from nominal.
4. Given the linear relation of step 2 and the set-point value specified in step 3, a *GPC controller can be synthesized using the original GPC algorithm* proposed by Clark [3]. Control parameters must be chosen appropriately to achieve satisfactory performance.
5. Similar to a traditional PSS, *the GPC PSS requires that its input be filtered by a washout filter* to prevent the controller from reacting to steady state changes in system operating point like those which can arise when governors act. A simple first order washout filter was used. Despite its necessity in the final design, however, the washout filter should not be considered in the synthesis process. The author’s experience is that doing so makes the stability of the synthesized controller erratic and selection of the control parameters thus becomes more difficult.
6. The validity of design should be evaluated by verifying that all the closed-loop modes are suitably damped and that the controller itself is stable.

Based on the steps enumerated above it appears as though the basics behind the design of a GPC PSS are relatively simple. Still, certain issues must be resolved before the design can be considered acceptable. The next three sections address these issues. One of the first problems encountered can be described as excessive controller complexity. It arises because the GPC algorithm will generally synthesize a controller having the same order as the model it is furnished with. Since power system models are generally very large and complicated a GPC controller synthesized using such models will be of very high order. This issue is addressed in section 3.2. The second task in designing the GPC PSS was to define a procedure for selecting the control parameters mentioned above in step four. This is a more daunting task than with traditional PSS design because the proposed design requires the selection of three parameters. These parameters are the *output horizon*, the *control horizon* and the *control penalty*. Thus there are three degrees of freedom as opposed to the one involved in conventional stabilizer design where only the stabilizer gain must be chosen. This problem is tackled in section 3.3. The last of the major challenges allocated a subsection of its own can be labeled controller offset elimination. Early designs of the GPC PSS produced steady state offsets in their output signals. Such an offset would disrupt normal voltage regulation and needed to be removed. This issue is addressed in section 3.4.

This section ends with a short discussion of two things that, though they do not warrant a section of their own, need to be addressed for the sake of completeness. The first is a problem that has been labeled *marginal instability* and the second concerns some of the implications of using a washout filter in the design.

Marginal Instability:

The aforementioned problem labeled marginal instability manifests itself as a closed-loop pole located very near the $|z| = 1$ stability boundary. This pole behaves erratically, fluctuating between states of marginal stability and marginal instability in an unpredictable fashion. Fluctuations between the two states of stability can be triggered during the design phase by very small changes in the chosen control parameters. Of more urgent concern, however, is that the fluctuations can also be triggered during operation by small changes in the operating point of the network.

The source of marginal instability can be traced back to the controller's integrator. Figure 3-1 shows a block diagram indicating the closed-loop process.

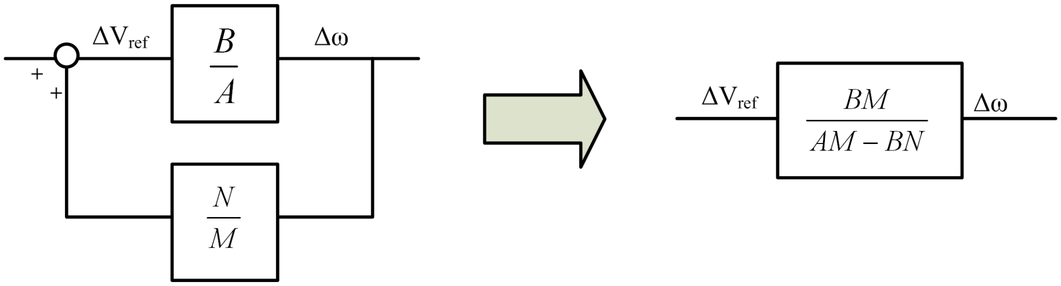

Figure 3-1: Diagram indicating the calculation of a closed-loop transfer function.

The basic GPC PSS will have a pole at z = 1 because of the integrator. If either B or N has a pole near z = 1 then so too will the closed-loop system. Consider it as pulling out a common factor from the polynomial products AM and BN. Determining whether the pole will be less than or greater than one is not easy to judge.

Although the instability discussed here will be small and has little noticeable effect over the times scales examined in this work it is not a good practice to have an unstable closed-loop process, regardless of how slight the instability. Fortunately, though, the problem can be addressed indirectly by some to the measures of section 3.4 which are used to eliminate the offset in the control signal.

General Implications of the Washout Filter:

Use of a washout filter to restrict the response of the GPC PSS to slow changes in operating point has a significant impact on the level of achievable performance. The filter's distortion of the machine speed deviation signal being fed to GPC algorithm appears to hamper the algorithms ability to regulate that signal. It should be noted that a washout filter has similar implications for the traditional PSS design.

The washout filter was also found to be strongly related to the problem of marginal instability because of the interaction of the controller integrator's pole at z =1 and the washout filter's zero at z = 1. (see conversion of washout filter from discrete-time to continuous-time domain shown in appendix B)

3.2 Controller Complexity

A controller designed using the GPC algorithm is generally of the same order as the LTI plant used in its synthesis. This presents a problem for power system applications because accurate network models are complicated and of very high order. Applying this complicated model directly to the GPC algorithm results in a very complicated controller. From a practical perspective this approach is inappropriate. It is both undesirable and unprecedented to incorporate such a controller into a power network and thus simplifications become necessary. One work-around is to synthesize the controller using a reduced-order model of the process, produced only from those dominant modes of the process that are problematic. The validity of such an approach is explored in this section. The process of model simplification is first outlined and then the reduced-order GPC controller is compared to the full-order GPC controller. A short discussion of the results follows.

3.2.1 Process Model Simplification

Assume for a moment that the transfer function corresponding to the relation of interest is written as shown in equation 3-1 by using partial fraction expansion.

$$\begin{aligned} H(z) &= \frac{b_o + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{a_o + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \\ H(z) &= \frac{R_1}{1 - p_1 z^{-1}} + \frac{R_2}{1 - p_2 z^{-1}} + \quad \cdots \quad + \frac{R_N}{1 - p_N z^{-1}} \end{aligned} \qquad \textbf{[3-1]}$$

Here, the R_n and p_n variables respectively represent the residues and the poles of the relationship. A simplified transfer function *H'(z)* can be constructed by retaining only those terms corresponding to the dominant modes. The aim is to mimic the behavior of *H(z)* using the reduced-order model, *H'(z)*. For instance if poles p_1 and p_2 of equation 3-1 were found to be dominant, then *H'(z)* would be written as in equation 3-2.

$$H'(z) = \frac{R_1}{1 - p_1 z^{-1}} + \frac{R_2}{1 - p_2 z^{-1}} \qquad \textbf{[3-2]}$$

For power system applications the dominant mode of the system will often be complex. In fact, a PSS is designed specifically to address problematic complex/oscillatory modes. If p_1 is complex then R_1 will generally be complex. Also, because the process in question is an actual physical plant it will necessarily have another pole p_1* with residue R_1*, where the asterisk indicates a complex conjugate. A simplified transfer function for a single complex mode can be written as given in equation 3-3.

$$H'(z) = \frac{R_1}{1 - p_1 z^{-1}} + \frac{R_1^*}{1 - p_1^* z^{-1}} \qquad \textbf{[3-3]}$$

Since the GPC algorithm given in [3] requires a difference equation in order to synthesize a controller, *H'(z)* must be written as a ratio of polynomials, B'(z)/A'(z). The two first order terms of equation 3-3 can be combined as illustrated in equation 3-4.

$$H'(z) = \frac{R_1 + R_1^* - \left(p_1^* R_1 + p_1 R_1^*\right) z^{-1}}{1 - \left(p_1 + p_1^*\right) z^{-1} + p_1 p_1^* z^{-2}} \qquad \textbf{[3-4]}$$

with

$$R_1 = R_R + iR_I$$
$$p_1 = p_R + ip_I$$

H'(z) can then be simplified such that all complex values are eliminated. The simplified equation is given in equation 3-5.

$$H'(z) = \frac{2 \cdot R_R - 2 \cdot \left(p_R R_R + p_I R_I\right) z^{-1}}{1 - 2 \cdot p_R z^{-1} + \left(p_R^{\,2} + p_I^{\,2}\right) z^{-2}} = \frac{b_0^{'} + b_1^{'} z^{-1}}{a_0^{'} + a_1^{'} z^{-1} + a_2^{'} z^{-2}} \qquad \textbf{[3-5]}$$

This transfer function can then be easily written as a difference equation and used directly in the synthesis of a GPC controller.

3.2.2 Reduced-Order GPC vs Full-Order GPC

When designing the GPC controller, use of the reduced-order model in place of the full-order model did not have a significant effect on the performance ceiling. Sometimes use of the reduced-order model in synthesizing a GPC controller enabled better performance and other times it was the use of the full-order model that would lead to better performance. Either way, the achievable damping for both approaches was usually comparable.

Next, the full-order and reduced-order process models were used to design two GPC controllers that are respectively referred to as the full-order GPC and the reduced-order GPC. For each, the controller parameters were selected, according to the guidelines given in the next section, to achieve approximately 20% damping for the dominant oscillatory mode. Although 5% would have been a more practical target, that level of damping was easily achieved with the OMIB test systems used. The intent of using a 20% target was to push the limits of the algorithm somewhat. What follows are the results from a comparison of the two controllers. It must be stressed that the lone purpose of performing the comparison at this junction of the report is to evaluate the viability of using a reduced-order model. Only the control penalty was varied to achieve the damping objectives, leaving all other parameters the same for both the full-order and reduced-order controllers. Controller saturation is neglected so as not to obscure results. In order to facilitate a more thorough evaluation the investigation is repeated twice, once for each of the test systems detailed in appendix A.

Test System 1

Figure 3-2 compares the machine speed deviation and GPC control signals for closed-loop systems employing the designed full and reduced-order GPC controllers. The disturbance applied at time t = 1s is a 0.1 pu step increase in the machine's mechanical torque input. The responses for the two closed loop processes are similar and both provide good regulation of the machine speed.

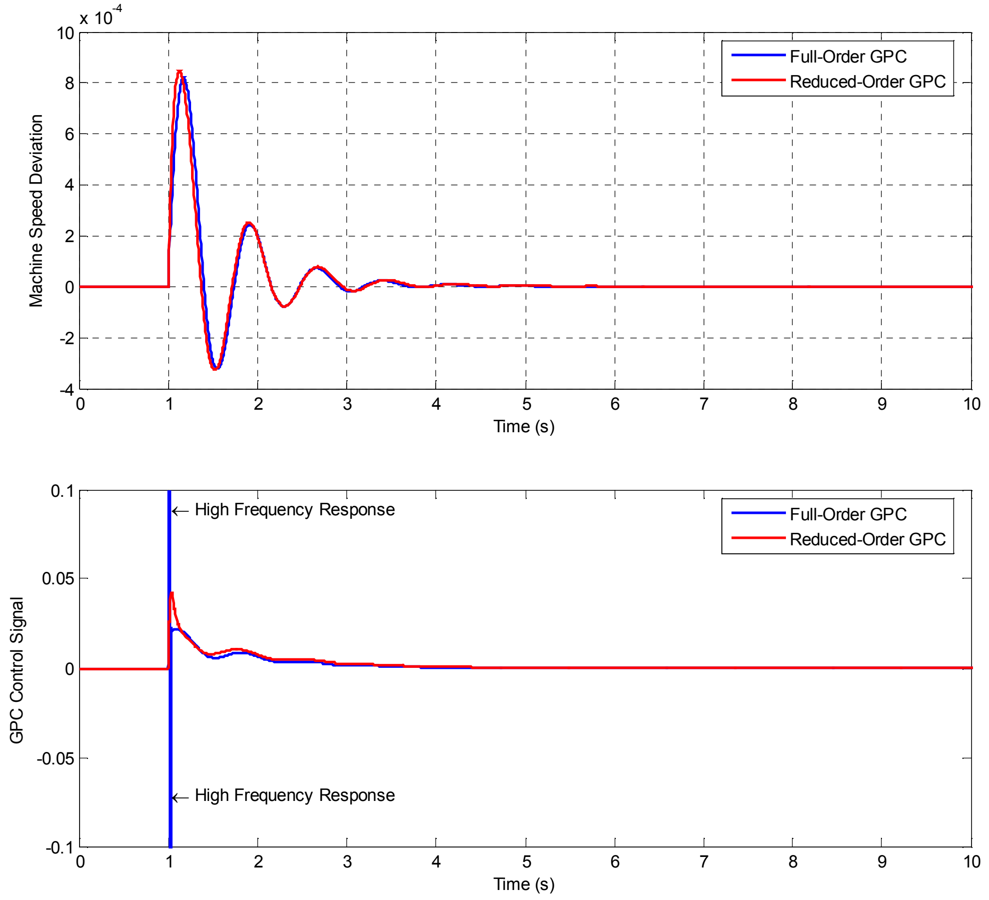

Figure 3-2: Comparison of machine speed deviation and GPC control signals from closed-loop processes employing the full-order and reduced-order GPC controllers. *(Test System 1)*

Based on the large initial transients in its control signal, the full-order GPC appears to provide more active control. The figure only shows a small portion of the actual transient. Such transients are absent from the controller response of the reduced-order GPC controller. The reduced activity of the reduced-order model is likely a direct consequence of using a simplified process model that does not consider the faster dynamics of the process. Still, despite the initial difference in response, the performance of the two controllers does converge after a cycle or so.

Next is an examination of the impact that use of the reduced-order model has on the robustness of the GPC controller during off-nominal operation. Four different parameters from Test System 1 were varied individually from their nominal values. The designed full and reduced-order controllers were then applied to linearized models of these off-nominal systems and the trajectories of the dominant closed-loop modes are plotted against each other in figure 3-3. Erratically behaved, marginally unstable modes are omitted in these series of plots. For the moment the reader should note that the marginally unstable modes are not a serious concern and that redress of the problem will be left for latter. The trajectories for the full and reduced-order controllers intersect at least once at the point corresponding to the nominal operating condition around which the controllers were designed. Neither design demonstrates conclusive evidence of superior performance in regards to robustness against the changing operating conditions. Sometimes the reduced-order model would provide better performance; other times it is the full-order controller. Still, for off-nominal operating points where the damping would fall below the design target of 20%, the full-order controller usually provided better performance than the reduced-order design.

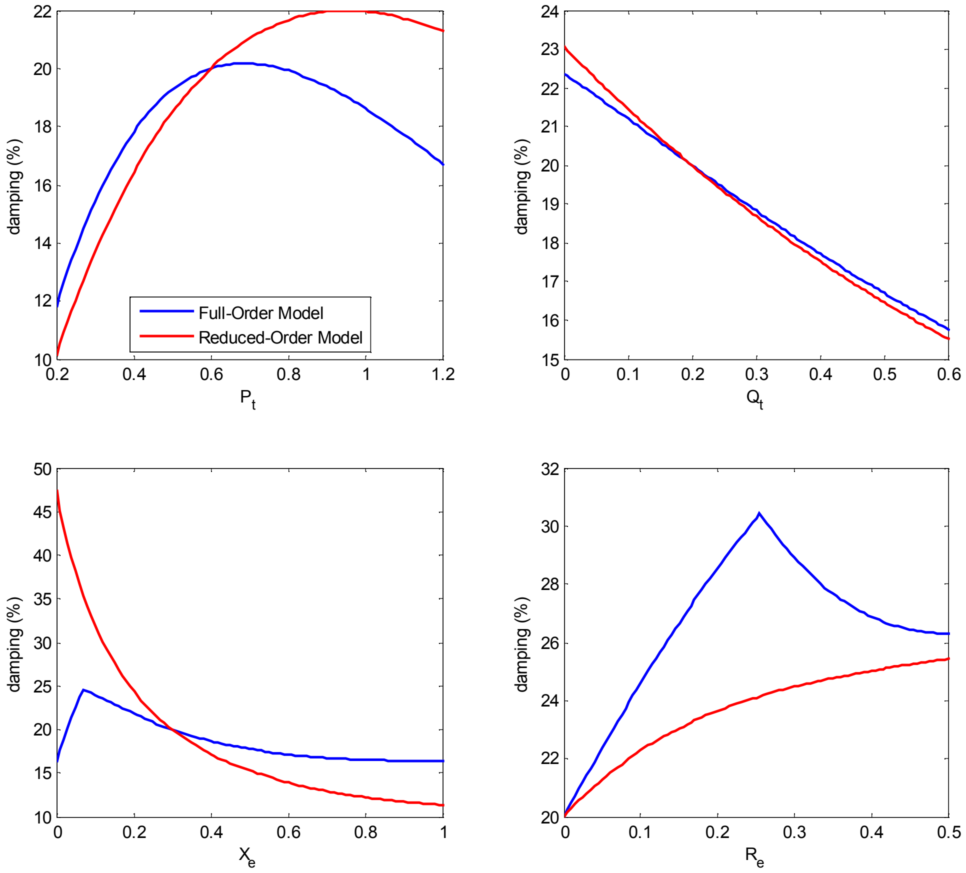

Figure 3-3: Trajectories of the dominant closed-loop modes resulting from application of the reduced and the full-order GPC controllers to various off-nominal operating conditions. *(Test System 1)*

Test System 2

As with Test System 1, a comparison of the machine speed deviation and the GPC control signals from the full and reduced-order controllers is given in figure 3-4. The disturbance is once again a 0.1 pu step increase in the machine's mechanical torque input and is applied at time t = 1s.

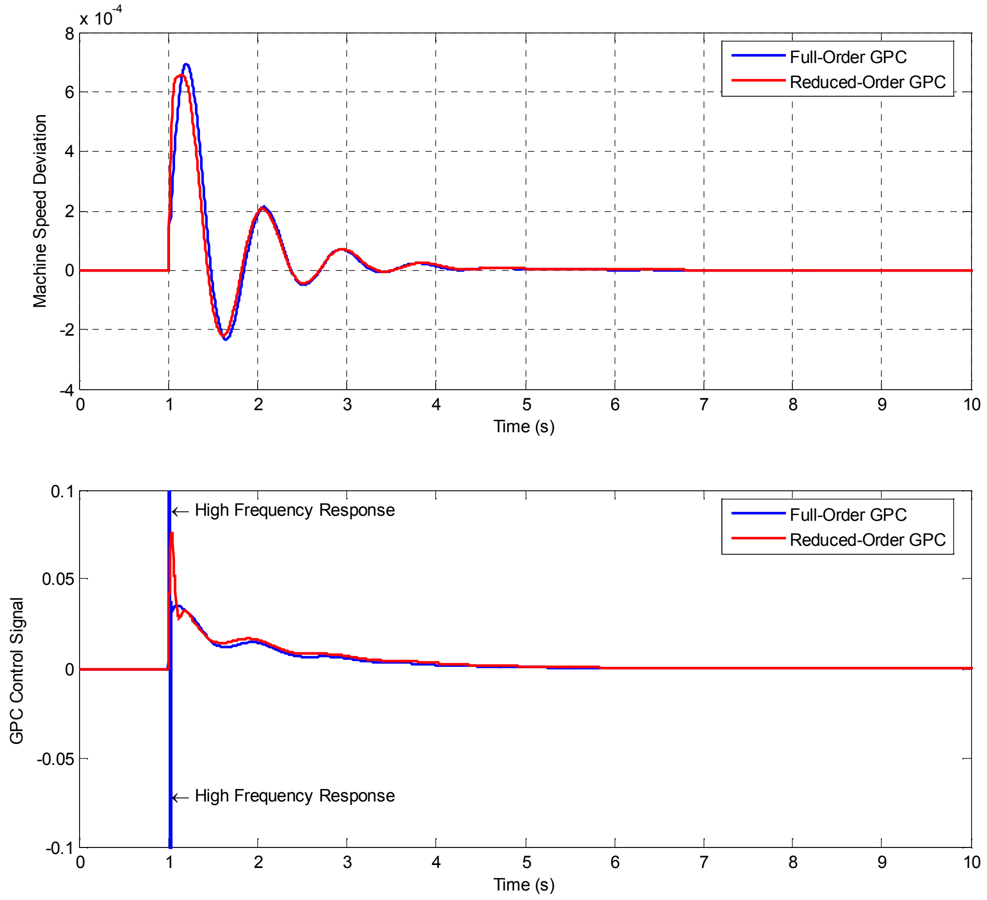

Figure 3-4: Comparison of machine speed deviation and the control signals from closed-loop processes employing the full-order and reduced-order GPC controllers. ***(Test System 2)***

Like before, both controllers provide very similar performance. As was the case with Test System 1, the full-order controller exhibited large initial transients in the control signal but the responses of two closed-loop systems converged after about a cycle of the low frequency oscillation.

Controller robustness was again examined by independently varying a number of the OMIB parameters to generate various off-nominal operating points. The designed controllers were then applied to linearized models representing these off-nominal operating points. Damping profiles for the dominant closed-loop mode are presented in figure 3-5 and are similar to those observed using Test System 1. Again, those modes that were marginally unstable are not considered. Neither the full-order controller nor the reduced-order controller demonstrated remarkably better performance than the other. Like before, however, when operating point changes caused damping to fall below the design target, the full-order GPC generally provided superior performance than the reduced-order GPC.

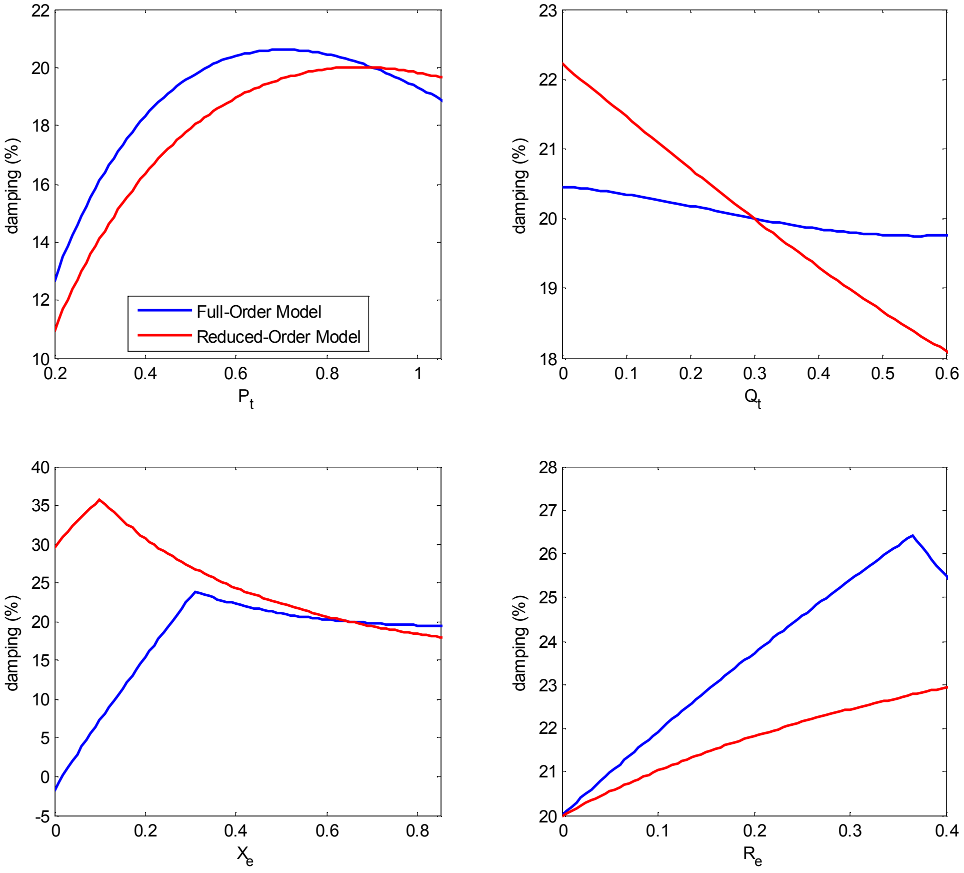

Figure 3-5: Trajectories of the dominant closed-loop modes resulting from application of the reduced and the full-order GPC controllers to various off-nominal operating conditions. *(Test System 2)*

3.2.3 Discussion

In this section it was shown that good closed-loop performance could still be achieved despite the use of a simplified process model in synthesizing a GPC controller. There was no marked difference in the level of achievable damping that was possible using the two designs. When a full-order GPC and a reduced-order GPC were designed to achieve the same level of damping their performance was very similar. The full-order controller was found to be slightly more robust to changing operating points while the reduced-order controller was found to have a preferable initial response to disturbances. In sum, the results presented here seem to support the use of a reduced-order GPC as a suitable substitute for a full-order GPC. It offers both some minor advantages and entails some minor drawbacks but has demonstrated that it can still lead to effective control. The bulk of the work that follows will use a reduced-order model of the process to synthesize the GPC controller because it is the only practical option; especially given the size and complexity of power networks.

3.3 Controller Parameters

Before a procedure for selecting the control parameters could be established, some understanding of how those parameters impacted the process being controlled was needed. The first subsection summarizes the observations made about the connection between the controller's parameters and the properties of the closed-loop process. Control horizon, output horizon and control penalty are the specific control parameters of interest. This is followed by an outline and justification of the procedure used throughout this work to select the control parameters in question.

3.3.1 General Observations

General observations pertaining to the impact of control and output horizons were made by repeatedly synthesizing the GPC PSS for different combinations of these parameters. In each case the damping of the dominant oscillatory mode was chosen near optimum by the appropriate selection of the control penalty. Although the previous section established the viability of using the reduced-order controller in place of the full-order controller, using both here is instructive. Mappings indicating the best case dominant mode damping as a function of the output and control horizons are presented in the plots of figure 3-6. Test System 2 was used throughout. No explicit caution was taken to ensure the practicality of each set of parameters in regards to controller instability. Marginally unstable poles are not considered.

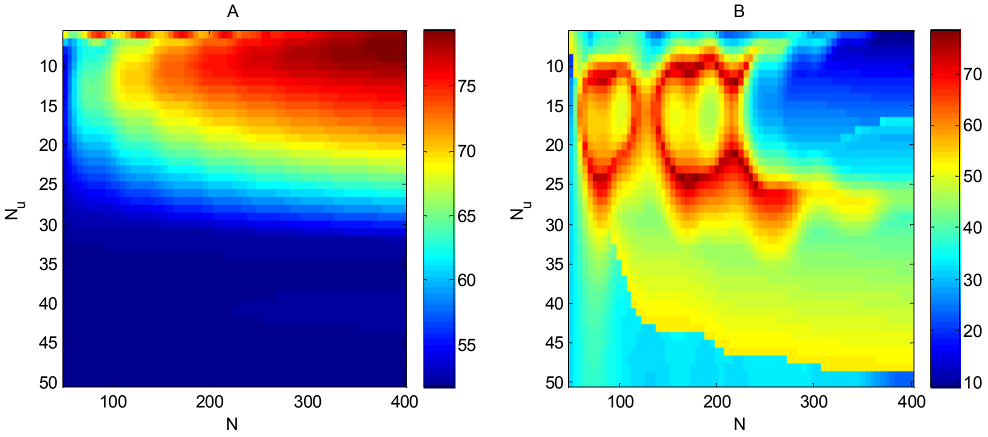

Figure 3-6: Mappings of achievable closed-loop damping as a function of the output horizon, N, and the control horizon, N_u. (A) Full-Order GPC and (B) Reduced-Order GPC.

Observations regarding the impact of control penalty were made using similar empirical means. This time the plots of figure 3-7 were used. For an arbitrarily specified combination of control and output horizon, the control penalty was varied over a range of values and the trajectory of the dominant mode's damping was plotted. Results are again

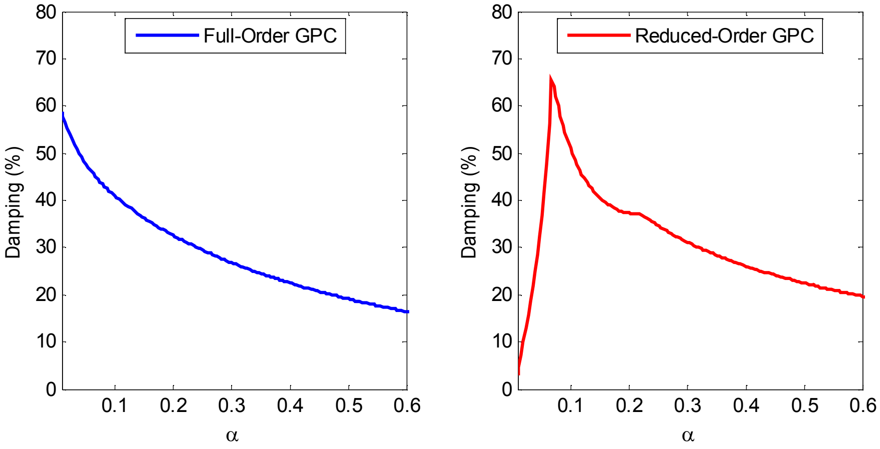

Figure 3-7: Damping trajectories of the dominant closed-loop mode which results from application of the full and reduced-order GPC controllers. (N = 50, N_u = 15)

gathered for both the full and reduced-order controllers. Again, Test System 2 was used and the stability of the controllers was not verified.

In the following, the examination of each control parameter begins with a discussion of what should be expected and ends with a description of what was generally observed by referring back to figures 3-6 and 3-7. Attempts to justify what was observed are made.

Output Horizon

When the assumed model coincides precisely with the actual model it is reasonable to expect that increasing the output horizon, N, would increase the ceiling on the level of achievable performance. This expectation follows from the first term of the GPC cost function, $\sum_{j=1}^{N}[y(t+j)-w]^2$. This term is the core of the GPC algorithm and leads to the selection of *Δu(t)* such that the succeeding N output predictions deviate as little as possible from the chosen set-point value, *w*. Given the choice, it seems intuitively preferable to have the controller act in such a way that the desired objective is achieved over the longest interval possible. Selecting the output horizon, N, large ensures this. Contrarily, too large an output horizon is expected to compromise performance in cases where the model used during controller synthesis fails to represent the actual process with sufficient accuracy. This failure of the model occurs to different degrees, for example, when the reduced-order model is used to approximate the process and during off-nominal operation. When the model is incorrect its long-term predictions become increasingly unreliable the further into the future they are. Deriving a control signal based on erroneous predictions increases the likelihood of inappropriate control action and thus is expected to lead to a less effective and less predictable controller.

Mapping A of figure 3-6 makes use of the full-order process model when synthesizing the GPC controller and is an example of when the assumed model coincides with the actual model. As expected, irrespective of the particular control horizon chosen, the maximum achievable damping grew monotonically with the output horizon. Mapping B uses the reduced-order process model when synthesizing the GPC controller and is an example of when the model used for controller synthesis deviates from the actual model. Focusing on the upper right hand corner of the plot where large control horizons do not dampen much the effect of a poor model, it becomes clear that performance suffers when the output horizon is large. For off-nominal operating conditions, yet another example of when the design model and actual model do not coincide, preliminary investigations showed that the range of stable operating points was reduced when the output horizon was extended excessively.

Control Horizon

When the assumed model coincides with the actual process model it is expected that the achievable performance should grow as the control horizon shrinks. Reasoning for such an expectation follows once again from the GPC algorithm's cost function. As mentioned above, the key component of the cost function is the first term. The second term only serves to limit the size of the control increment and does not contribute directly to our objective of minimizing the deviation between the predicted and desired output. Therefore, if the impact of the second term in the cost function is reduced then one can expect better regulation of the output. Reduction of the control horizon accomplishes this. At the other end of the spectrum, when the process model used in controller synthesis is in error, having a small control horizon is a liability. With a small control horizon, the GPC algorithm relies too heavily on the process model for the derivation of an appropriate control signal. Like before using an inaccurate model to derive a control signal will lead to the increased likelihood of improper control.

For mapping A of figure 3-6 where the design model and actual model were the same, decreasing the control horizon led rather consistently to an increase in the achievable performance. For mapping B of the same figure where the design model differs from the actual model, the simple relation between control horizon and the achievable damping is lost. No pattern is easily discernible. During off-nominal operation, increasing the control horizon seemed to degrade performance. This latter result is both unexpected and unexplained since increasing the control horizon should have a neutral or slightly beneficial impact on robustness. A larger control horizon should lead to less dependence on the accuracy of the design model and thus should result in a controller that is more robust to varying operating points.

Control Penalty

A reduction in the control penalty should make calculation of the control signal more dependent on the process model. In the ideal case where the process model used during controller synthesis coincides with the actual model, this increased dependence should lead to more active and thus generally more effective control. An increase in control penalty should do the opposite. The heavy reliance on the design model, which is precipitated by a small control penalty, is expected to become a liability whenever that model does not match the process actually being controlled. Like before, deriving controls largely on the basis of an incorrect model increases the likelihood of poor performance. Finally, irrespective of how well the design and actual models coincide, controllers with very large control penalties are expected to be ineffective, exerting little influence on the open-loop processes they control.

In ideal conditions where the design and actual process models coincide, the GPC controller generally responded to a change in control penalty in the expected manner. For the most part performance increased as the control penalty decreased. One of the plots of figure 3-7 corresponds to when a full-order GPC controller is employed and illustrates this trend. When there was a discrepancy between the design model and the actual model a new trend emerged. Decreasing the control penalty could not generally be relied on as a means of increasing performance. The second plot of figure 3-7 corresponds to the use of a reduced-order GPC controller and shows that decreasing the control penalty increases performance only up to a certain point. Further reductions in control penalty then degraded performance.

3.3.2 Control Parameter Selection

Now that some observations have been made regarding how the control parameters affect performance a procedure for selecting them can be established. An important thing to note from the plots of figure 3-6 is that while certain combinations of the output and control horizons can lead to improved damping over other combinations, almost any is capable of providing good damping so long as the control penalty is appropriately selected. Given this reality, an intuitive approach rather than an optimum one is adopted to select the control parameters.

Output Horizon Selection

The designed GPC controller is intended to damp LFOs that range in frequency from 0.5 to 3 Hz. With a 0.01s sampling period, an output horizon of N = 50 implies that the optimization of the control objective will be carried out using the process model's predicted output for the next 0.5 seconds. Making control decisions based on predictions of the process output over the coming 0.5 second interval does not seem overly presumptuous. Simple investigations show that increasing the output horizon much beyond that entails degradation in the controller's abilities to cope with non-ideal operating conditions. For these reasons, the work which follows will generally use an output horizon of N = 50, straying only when performance requirements necessitate doing so.

Control Horizon and Control Penalty Selection

Selection of the control horizon and control penalty is made via trial and error. First, several candidate values of control horizon are selected, bearing in mind the general observations from the previous subsection. For each pairing of the chosen output horizon with a candidate control horizon, a plot can be constructed which shows how the magnitude of the closed-loop poles vary with control penalty. Likewise, the damping of the closed-loop's dominant mode can also be plotted as a function of control penalty, α. Examples of the discussed plots are provided in figure 3-8. Only oscillatory modes are considered in the damping trajectory shown. A plot of the magnitude of the controller's dominant mode as a function of control penalty is also important in making good control parameter selections. Such a plot enables a designer to verify that a specific set of control parameters lead to a stable controller. Figure 3-9 gives an example of the aforementioned plot. The range of control penalties used is smaller than in figure 3-8 because it is for smaller control penalty values that controller instability tends to occur.

Equipped with the plots described above, a designer can then chose a set of control parameters which lead to the desired damping performance. Suitable control parameter choices must avoid the controller stability limit indicated in figure 3-9. A designer should also strive to avoid control parameter choices that lead to damping trajectories with certain undesirable characteristics. Choices of control horizon that lead to trajectories with abrupt peaks in the damping trajectory of the dominant mode should be avoided. Likewise, choices of control horizon that cause the stability of the closed-loop system to become erratic and strongly dependent on the choice of control penalty should be avoided. When selecting a control penalty, there will generally be two values that lead to the desired damping performance, one on either side of the peak. The control penalty should always be chosen as large as possible. Too small a control penalty will change the dominant mode's frequency in addition to its damping and should be avoided.

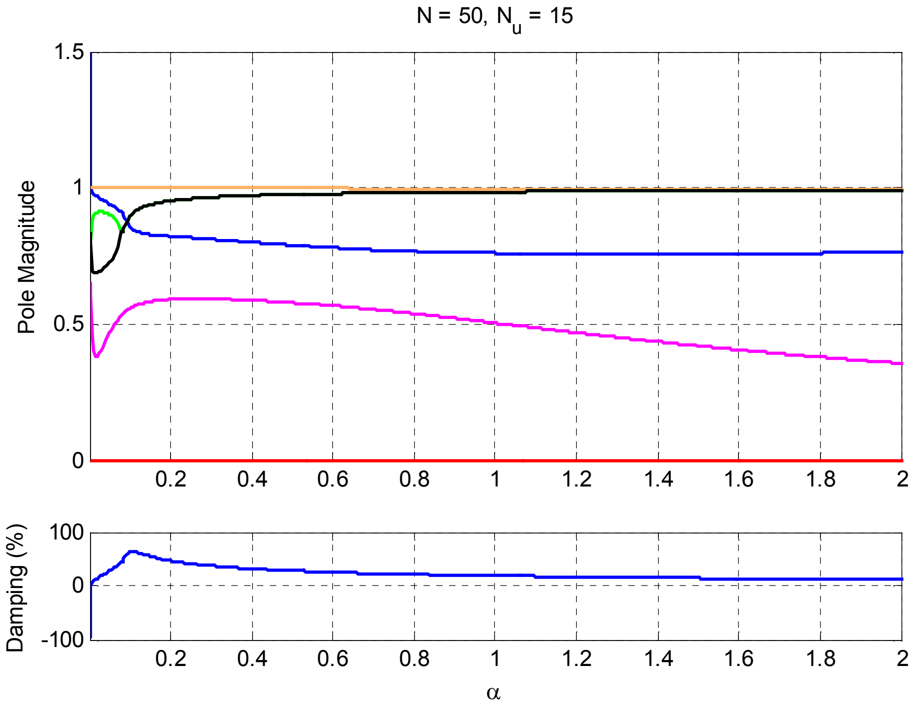

Figure 3-8: Trajectories of closed-loop poles for a GPC controlled process and the trajectory of the damping for the closed-loop's dominant oscillatory mode.

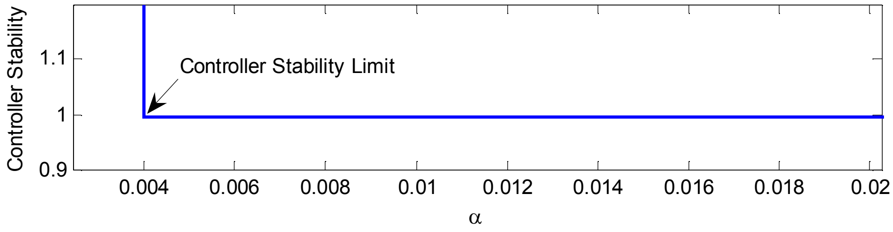

Figure 3-9: Stability of the GPC controller as a function of the control penalty, α.

3.4 Controller Offset Elimination

Clarke's original paper [3] proposing the GPC algorithm employs an integrator at the output of the controller. This integrator presents a particular challenge for the PSS control problem because it makes the presence of a steady state offset in the controller output signal a possibility. Such an offset could interfere with the voltage regulator of the local generator.

The basic GPC PSS design described earlier in this chapter includes a washout filter to limit the controller's response to persistent steady state changes in operating point. The use of the washout filter conveniently provides an ad hoc solution to the controller offset problem. It can be illustrated that eliminating the offset in the signal fed to the GPC algorithm will eliminate the offset in its steady state output. Pre-filtering the input to the GPC algorithm using a high pass filter such as a washout, while not completely eliminating its offset, does significantly reduce it. This in turn leads to a significant reduction in the size of the controller output's offset.

Because of the importance of the matter, it was decided that a systematic and specific means of eliminating the controller offset was preferable to relying solely on the ad-hoc solution furnished by the washout filter. This section details three different approaches. They include using a composite input signal, the removal of the integrator and finally using a modified integrator. Ultimately, modifying the integrator proved to be the best option and provided some additional benefits in terms of robustness. All of the simulations that follow are carried out on Test System 2, the more volatile of the two test systems. In order to avoid obscuring the results, the washout filter is omitted for the most part from the investigations that follow. A short discussion regarding the implications of including it will be given where appropriate because it is a practically necessary component of the design. Controller saturation is likewise omitted.

3.4.1 Composite Input Signal

The first attempt to eliminate the control signal offset was made by adding to the GPC's input an error signal proportional to the deviation of the machine's terminal voltage from nominal. Rather than have the GPC algorithm drive the $\Delta\omega$ signal to zero, a new signal $M = C_1 \cdot \Delta\omega + C_2 \cdot \Delta E_t$ is defined and the GPC algorithm drives it to zero. With the AVR regulating the terminal voltage, inclusion of ΔE_t term in the cost function was expected to cause the output of the GPC PSS to become zero. The belief was that zero was the only viable steady state value because a non-zero value would offset the control action of the AVR, leading to a non-zero value for ΔE_t and thus to failure of the controller to meet its control objective.

Selection of the weighting constants C_1 and C_2 was done by trial and error. In order to ensure that the PSS was not transformed into a voltage regulator, the contribution of $C_1 \cdot \Delta\omega$ to M was made greater than that of $C_2 \cdot \Delta E_t$. It was discovered that an effective controller leading to a stable closed-loop process could only be synthesized when the weighting constants C_1 and C_2 where given opposite polarities in what amounts to a difference of weighted signals as opposed to a sum thereof. Principally there is little difference in philosophy whether a difference or sum is used for the composite input. For either case it is well established that after a disturbance $\Delta\omega$ will tend towards zero in the steady state because it is the derivative of the state variable $\Delta\delta$ which will, by definition of the steady state, have settled to some constant value. With either a sum or a difference the objective of having zero terminal voltage deviation follows naturally from efforts to minimize the value of M since the machine speed deviation tends to zero. It simply appears to be the case that a difference is more appropriate for stability purposes than a sum. The work that follows will assume $C_1 = 1$ unless otherwise noted.

Nominal Operating Point Performance

The time domain responses of four different closed-loop processes employing four different controllers are shown in figure 3-10. The first doesn't use a composite input and is provided for comparison purposes. The other three use a composite input signal, each with different weighting constants. All four controllers were designed to provide the closed-loop process with ~20% damping. Increasing the magnitude of the weighting constant C_2 much beyond 1/50 led to an unstable closed-loop process. A step increase of 0.1 pu in the machine's mechanical torque input, which effectively amounts to a change in operating point, is applied at time t = 5s.

Examining first the case that doesn't make use of a composite input signal, the problem presented by the integrator becomes clear. The GPC PSS generates a steady state offset in its output while it attempts to damp out the LFOs. The size of this offset will generally depend on the nature of the disturbance. For instance, a large persistent disturbance might generate a large offset while a small short disturbance might generate a negligibly small offset. In the steady state, the aforementioned offset has to be compensated for by the AVR as it attempts to regulate the terminal voltage. The undesirable situation thereby develops where one controller ends up compensating for the deficiencies of another.

Moving on to the closed-loop systems employing a composite input controller, it appears as though its use has only a small impact on the controller's speed regulation abilities. Meanwhile, its use augmented regulation of the machine's terminal voltage, reducing the small persistent offset to zero. The speed of regulation increased as the weighting constant, C_2, increased. Although the introduction of the composite input did result in a significant reduction in the offset of the control signal it did not reduce it to zero as expected. Instead, it seems to have caused the reduction of the AVR control signal to zero. It thus appears as though use of the composite input leads the GPC PSS to monopolize voltage regulation responsibilities. This result implies that the logical basis for the composite input GPC was flawed. Its failure is due to a lack of consideration given to how the AVR and the GPC PSS interact. The implications of the GPC's faster response time were not considered. Presumably acting faster than the AVR, the GPC PSS preempts it, and appropriates its function.

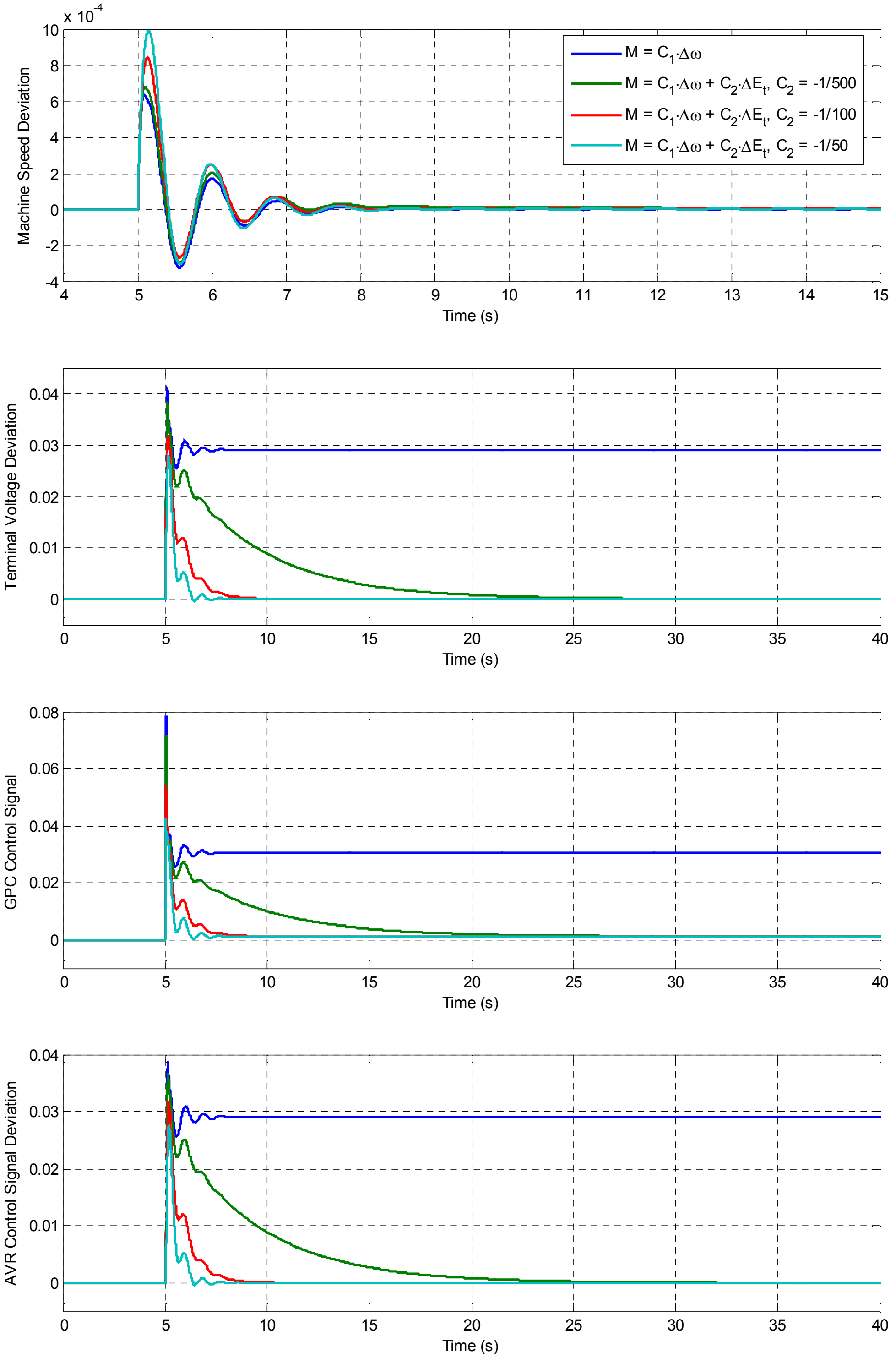

Figure 3-10: Time-domain responses of several composite-input GPC-controlled processes when the machine is subjected to a 0.1 pu step increase in its mechanical torque input.

Off-Nominal Operating Point Performance

Figure 3-11 shows the trajectories of the dominant closed-loop mode's damping for a number of different off-nominal operating conditions. As before, these off-nominal conditions are generated by varying the indicated system parameter while keeping all the others at their nominal values. There was little apparent effect on robustness from the application of the composite input based controller. When the weighting constant was given a conservative value of -1/500, the off-nominal damping trajectory was virtually unchanged from when the composite input wasn't used. Even setting the weighting constant aggressively didn't trigger dramatic changes in the damping profile.

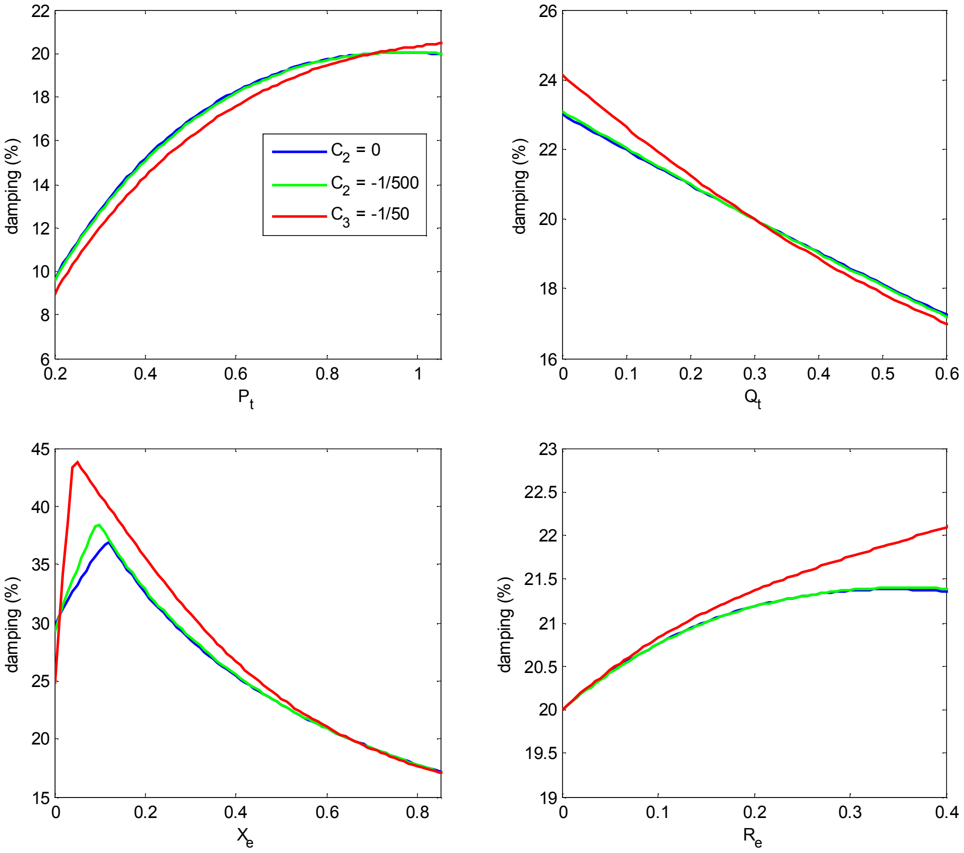

Figure 3-11: Dominant closed-loop mode damping trajectories which result when different composite-input GPC PSSs are subjected to varying off-nominal operating conditions.

When the range of operating points is expanded beyond the threshold where the open-loop properties of the process change dramatically as documented at the end of appendix A, it becomes obvious that introduction of the composite input signal causes the closed-loop process to undergo a dramatic and sudden instability. The instability that occurs is non-oscillatory in nature and its severity increases as the magnitude of the weighting constant, C_2, is increased. This is evident from the data of table 3-1 which shows how the magnitude of the dominant closed-loop mode at an arbitrary unstable operating point varies with the weighting constant C_2. The pattern of increasing instability as the magnitude of the C_2 rises holds for the five unstable operating points examined. Attaching too much importance to these observations, however, is unadvisable because the dramatic changes in open-loop system properties on which they depend are unlikely to be observed in any practical system. They are mentioned mainly for the sake of completeness.

C2	**Pt = 1.2**	**Qt = -0.3**	**Xe = 1.2**	**Re = 0.5**	**Et = 0.9**
0	0.999999999997	1.000000000032	1.000000000042	0.999999999945	1.000000000060
-1/500	1.008533393652	1.005421636292	1.006610391130	1.003187460994	1.007021730546
-1/100	1.028259969316	1.022733836550	1.022161356167	1.013870398004	1.024499916535
-1/50	1.041050002705	1.036886964778	1.031501398206	1.022655918710	1.035744030750

Table 3-1 : Evolution of five different unstable closed-loop modes as a function of the weighting constant C_2.

Washout Filter Impacts:

Use of the washout filter alone caused the steady state offset in the control signal to decay and addition of a composite input signal scheme for the GPC controller sped that decay. Unlike the case without the washout, however, the GPC PSS control signal did not completely preempt that of the AVR. Instead, as the magnitude of the weighting constant C_2 was increased, the GPC PSS would assume more of the voltage regulation responsibilities. This is illustrated by the data of table 3-2 which summarizes, as a function of the weighting constants, the contribution of the GPC PSS and AVR to the post-disturbance steady state deviation in the nominal feedback signal sent to the exciter.

C1	C2	GPC PSS	AVR
1	0	0.00%	100.00%
1	-1/500	24.78%	75.22%
1	-1/100	63.25%	36.75%
1	-1/50	78.52%	21.48%

Table 3-2: Contribution of the AVR and GPC PSS as a function of the weighting constants to the total steady-state exciter reference signal.

3.4.2 Integrator Removal

One paper that employed the GPC algorithm in the design of a PSS got around the control offset problem by removing the integrator. [6] Clearly this approach can provide a solution but it is not without its costs.

Nominal Operating Point Performance

Consider the selection of control parameters. A few interesting observations can be made by analyzing a sample of the closed-loop pole trajectories which result from the application of the GPC without integrator. Figure 3-12 furnishes such a sample. An important feature to note about the trajectory shown is that the selection of the control penalty is more critical than it is when an integrator is included. Good damping for the dominant mode occurs for a tighter range of control penalties than it otherwise would. Also, with the omission of the integrator, the control penalty has to be selected relatively small in order to achieve damping levels comparable to those obtained when an integrator is included. Intuitively this makes sense since the GPC algorithm optimizes the selection of the control increment. Without the integrator, what was once the control increment becomes the control signal. Naturally, the control increment must be more active and thus restrictions placed on it must be relaxed by reducing the control penalty. Because of the smaller control penalties, the separation is reduced between the control penalties that lead to the best performance and those whose choice leads to either closed-loop instability or controller instability. By comparison, when an integrator is included the optimum control penalties are well clear of those values that lead to instability.

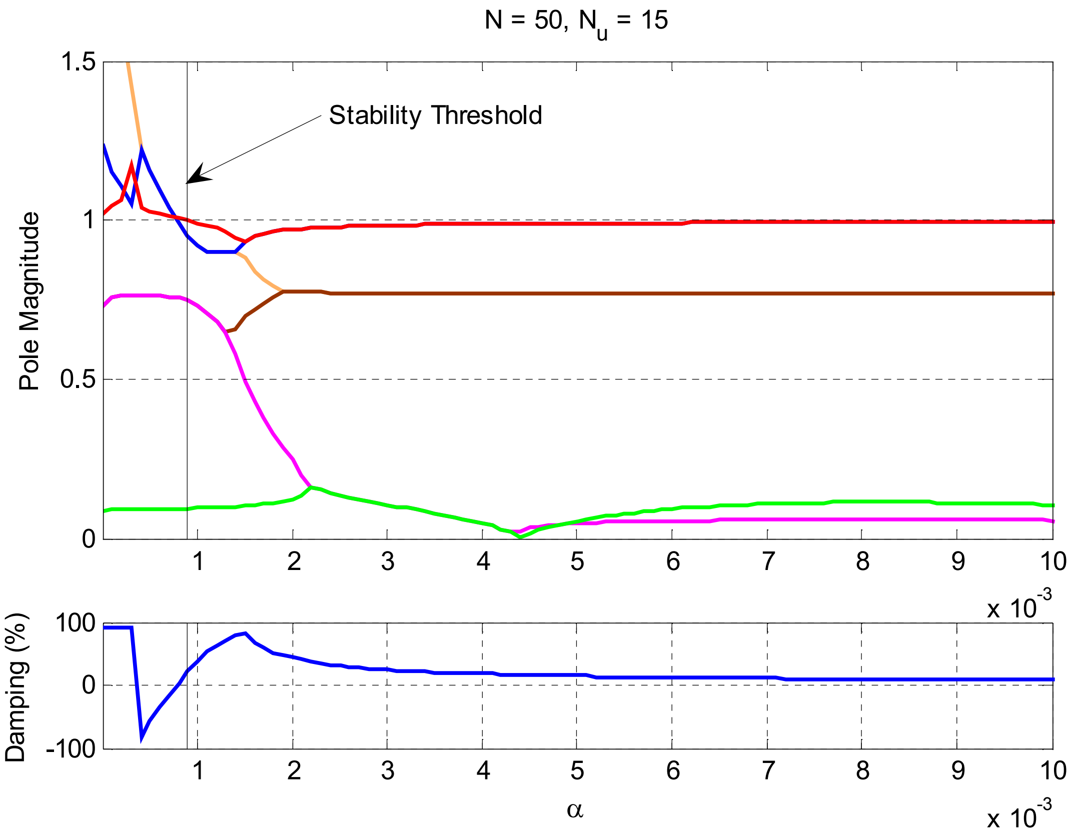

Figure 3-12: Trajectory of closed-loop poles resulting from application of GPC controller without an integrator. The damping of the dominant closed-loop mode is also shown.

Off-Nominal Operating Point Performance

Figure 3-13 shows how removal of the controller's integrator affects how the damping of the dominant closed-loop mode varies with operating point. Two controllers were synthesized, one with an integrator and the other without. Both are designed to achieve roughly 20% damping for the dominant mode at the nominal operating condition.

Examining these plots, it is difficult to draw conclusions. For the off-nominal operating point characterized by changes in either the machine's real or reactive power output, the two controllers perform comparably. For the off-nominal operating points characterized by changes in line impedance and resistance, the GPC controller without the integrator generally provides significantly better damping performance. Whenever the off-nominal operating conditions were such that they caused damping performance to dip below the design specification of 20%, the GPC controller with an integrator performed better, almost without exception.

Like in the previous subsection, removal of the integrator triggered abrupt failure of the closed-loop system when the range of off-nominal operating points was expanded to include those where the properties of the underlying process changed dramatically. Again, though, little emphasis should be placed on these observations since they are unlikely to be witnessed in a practical system.

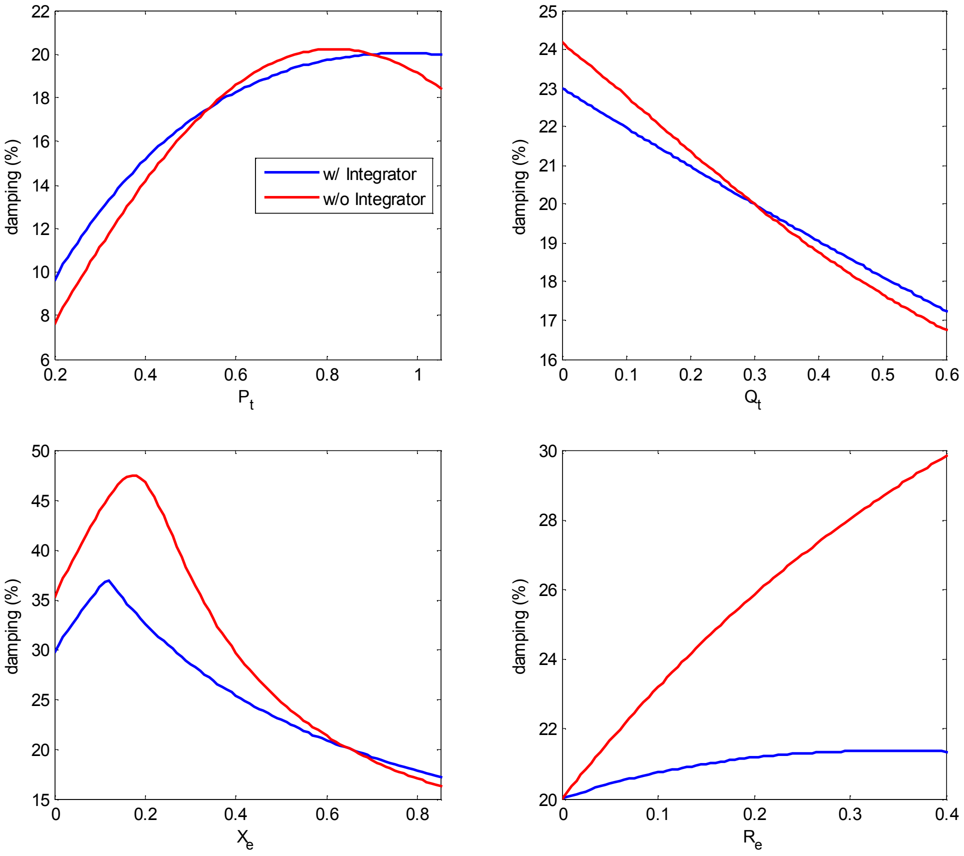

Figure 3-13: Trajectory of the damping of the closed-looped system's dominant mode under varying off-nominal operating conditions.

3.4.3 Lossy Integrator

The last attempt at a systematic approach to eliminating the offset in the control signal was made through modification of the GPC's integrator. Instead of a traditional integrator which can be expressed as $u(t) = u(t\text{-}1) + \Delta u(t)$, the equation $u(t) = \beta \cdot u(t\text{-}1) + \Delta u(t)$ was used. The modified integrator will be referred to as a *lossy integrator.* The constant β will be referred to as the *integrator loss factor* and is assigned a value that is both positive and less than unity. The introduction of β effectively gives the integrator a finite memory. The smaller its value, the faster the integrator's output should tend towards zero.

Nominal Operating Point Performance

Repeated time-domain simulations were carried out for various values of the integrator loss factor. The time responses of the machine speed deviation, control signal, terminal voltage deviation and AVR control signal deviation which result from these simulations are given in figure 3-14. All simulations use the same control parameters and achieve damping of ~20% for the dominant oscillatory closed-loop mode.

The first thing to note is that each controller's ability to regulate the machine speed is virtually unaltered by the use of a modified integrator. As expected, the smaller the value of the integrator loss factor, the faster the control signal decays to zero. The reduction of the controller offset to zero in the steady state meant that the AVR no longer had to compensate for it. The result was a significant reduction in the size of the steady-state AVR signal. Similarly, the steady-state offset in the machine terminal voltage was significantly reduced.

Off-Nominal Operating Point Performance

Figure 3-15 compares how the damping of the dominant oscillatory mode varies with operating point for different integrator loss factor values. A quick examination of the plots of this figure reveal that use of a lossy integrator has almost no effect on the robustness of the controller.

As with the other offset mitigating techniques examined, a controller employing a lossy integrator was unable to cope with off-nominal operating points that triggered a sudden change in the properties of the underlying process. Like before this observation is mentioned solely for the sake of completeness since real power system do not generally undergo such large and dramatic changes.

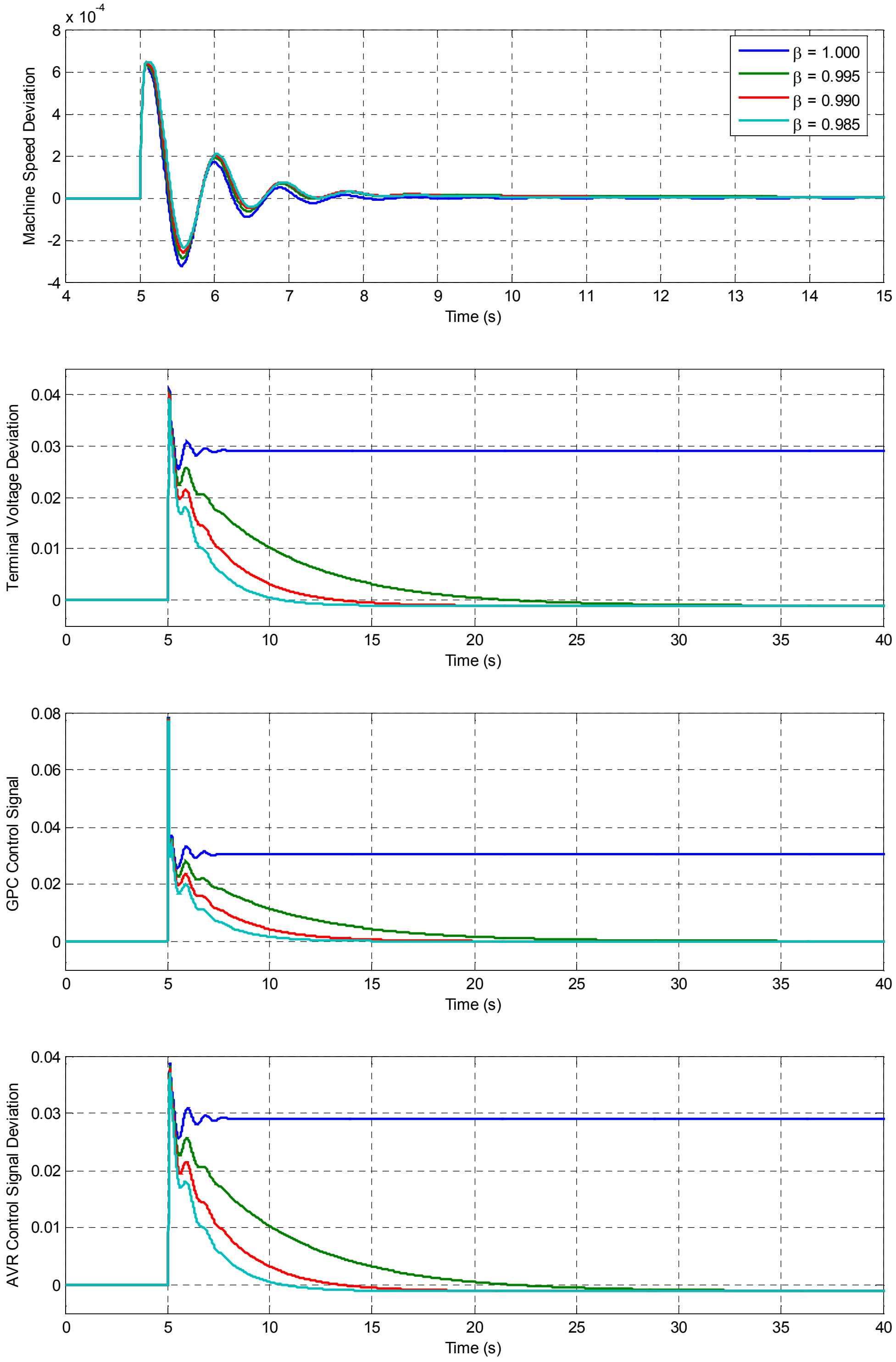

Figure 3-14: Time-domain responses of several lossy-integrator GPC-controlled processes when the machine is subjected to a 0.1 pu step increase in its mechanical torque input.

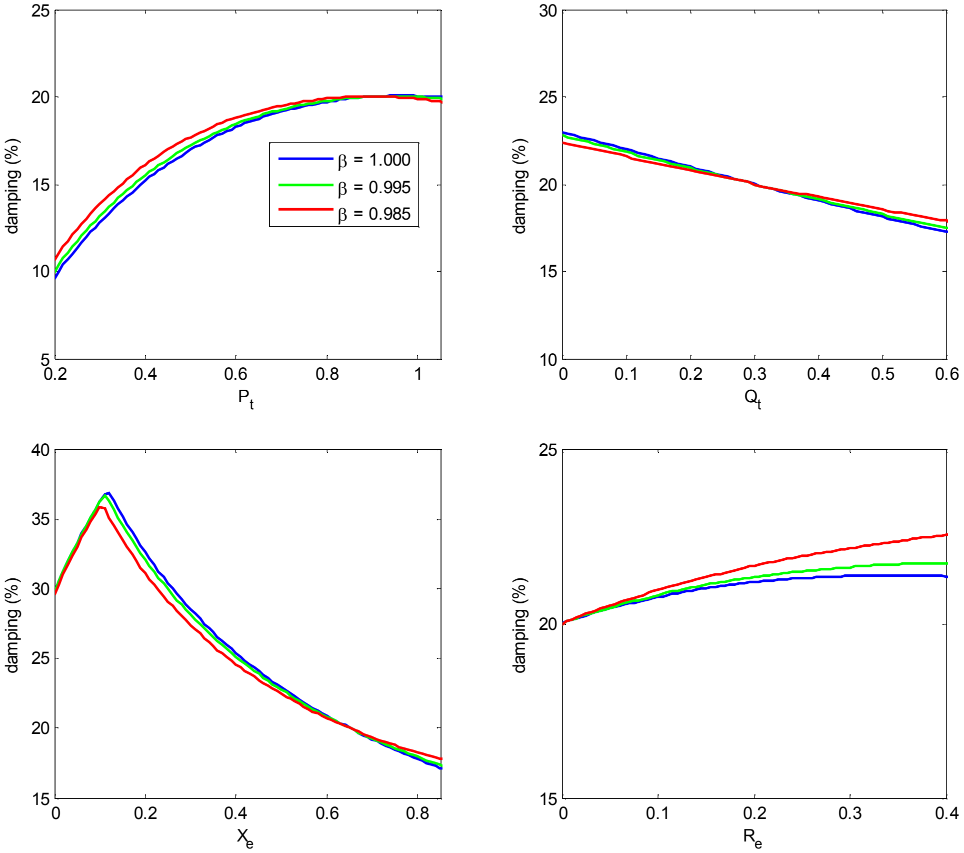

Figure 3-15: Trajectory of the damping of the closed-looped system's dominant mode under varying off-nominal operating conditions.

Washout Filter Impacts:

As far as the time domain response is concerned, the use of the lossy integrator in tandem with the washout filter sometimes led to slightly degraded regulation performance compared to that yielded by the washout alone. The control signal changed from being monotonic to being oscillatory.

3.4.4 Discussion

Although an ad hoc solution was provided through the use of the washout filter, a systematic approach was desired to address the issue of the offset generated by the integrator found at the output of the GPC controller. Although it is not necessary to eliminate the offset in order to achieve voltage regulation, the situation that arises from leaving it is undesirable. The AVR ends up compensating for the GPC PSS control signal. It is not a good practice to have one controller compensate for another. In the absence of a disturbance the GPC control signal should approach zero, leaving the AVR to function unabated.

The first attempt at eliminating the offset involved using a weighted sum of $\Delta\omega$ and ΔE_t as the input to the GPC algorithm instead of $\Delta\omega$ alone. One of the subtle benefits of this method is that it actually results in the regulation of the terminal voltage to zero. If the terminal voltage regulation duties were left solely to the AVR, a slight steady state offset would appear. The offset stems from deficiencies in the design of AVR. This approach, however, proved to be fundamentally flawed and the GPC PSS began usurping the function of the AVR. This was a recurrence of the original problem where one controller was compensating for another.

The second approach involved the removal of the integrator. It was appealing for two reasons. First, it allowed for the elimination of the control signal offset as intended. Secondly, it provided the added benefit that without the integrator, the problem of marginal instability vanished. These admirable properties were countered by two pitfalls. First, for operating points that triggered the performance to fall below the design target, the GPC PSS with an integrator seemed to fairly consistently lead to better performance than one without it. The second pitfall was that removing the integrator complicated the design process. Smaller control penalties requiring more judicious selection were needed to ensure an effective and stable closed-loop process. This latter point would be of increasing concern within the context of adaptive control, where self-tuning may be employed.

The final attempt was described as the use of a lossy integrator and involved modification of the original algorithm's integrator. The output of the GPC PSS under such a scheme decayed to zero in the steady state as desired. Also, the problem of marginal instability was resolved indirectly using this scheme. The controller had its pole moved from $z = 1$ to $z = \beta$ and thus the problem of poles interacting near the stability boundary was resolved. A final advantage is that use of the lossy integrator had only a marginal effect on performance during off-nominal operation and had virtually no impact on the selection of control parameters. Because of these enumerated advantages and because it is generally very un-invasive, the lossy integrator approach is adopted to eliminate the control signal offset in the work that follows.

3.5 GPC PSS vs Traditional PSS

Having established a design for the proposed GPC PSS, the next phase is to test it. An appropriate means of doing this is to compare it against a conventional PSS. To complete a thorough investigation, three different control problems were formulated and are listed below. Each of these control problems was resolved using both the traditional PSS and the proposed GPC PSS design approaches. All of these problems made use of Test System 1. Formulating similar problems using Test System 2 provides little new insight or understanding and is therefore omitted.

Control Problem #1:
For the operating point and configuration defined by Test System 1 of Appendix A, design a controller to ensure that all closed-loop modes have damping of at least 5%.

Control Problem #2:
For the operating point and configuration defined by Test System 1 of Appendix A, design a controller to ensure that all closed-loop modes have damping of at least 40%.

Control Problem #3:
For the operating point and configuration defined by Test System 1 of Appendix A, design a controller to maximize the damping of the closed-loop system's dominant mode.

Three different GPC PSSs were synthesized using the methods outlined thus far in this chapter, one for each of the control problems listed above. Likewise, three different traditional PSSs were designed. Designs corresponding to the same control problem are now compared. First performance at the nominal operating point is examined before proceeding with an investigation of off-nominal performance. Controller saturation is excluded from the simulations that follow in order to avoid obscuring results. The design of the traditional PSS and the GPC PSS controllers for *Control Problem #1* are presented as examples in appendices C and D respectively.

3.5.1 Nominal Operating Point Performance

Traditional PSS #1 vs GPC PSS #1

Both controllers were designed for the open-loop process labeled Test System 1 of appendix A and were intended to ensure a minimum of 5% damping on all the closed-loop process modes. The 5% target was a modest one and was not difficult to attain for an OMIB using either the traditional or GPC PSS. The ease with which this objective is obtained allows for the selection of fairly conservative parameter values. Table 3-3 summarizes the dominant modes of each closed-loop process. More complete listings of the closed-loop modes can be found in appendices C and D.

	Eigenvalues	|Eigenvalues|	Damping Ratio (%)	Frequency (Hz)
Traditional PSS #1	0.99270 ± 0.080529i	0.99596	4.9996	1.2899
GPC PSS #1	0.99269 ± 0.080586i	0.99596	4.9961	1.2908

Table 3-3: Dominant modes and their properties for closed-looped processes employing Traditional PSS #1 and GPC PSS #1 designs.

There is little difference between the dominant modes of the two closed-loop processes. The frequencies of oscillation are almost identical and differ only slightly from the open loop value of 1.3041Hz. Based on these similarities the time domain responses yielded by the two closed-loop systems are expected to be very similar. The plots of figure 3-16 provide such a comparison. The response of the open-loop process is also given for the sake of having a performance reference. The disturbance takes the form of a 0.1 pu step change to the machine's mechanical torque input and is applied at time t = 5s. The three signals shown are the machine speed deviation, terminal voltage deviation and controller output.

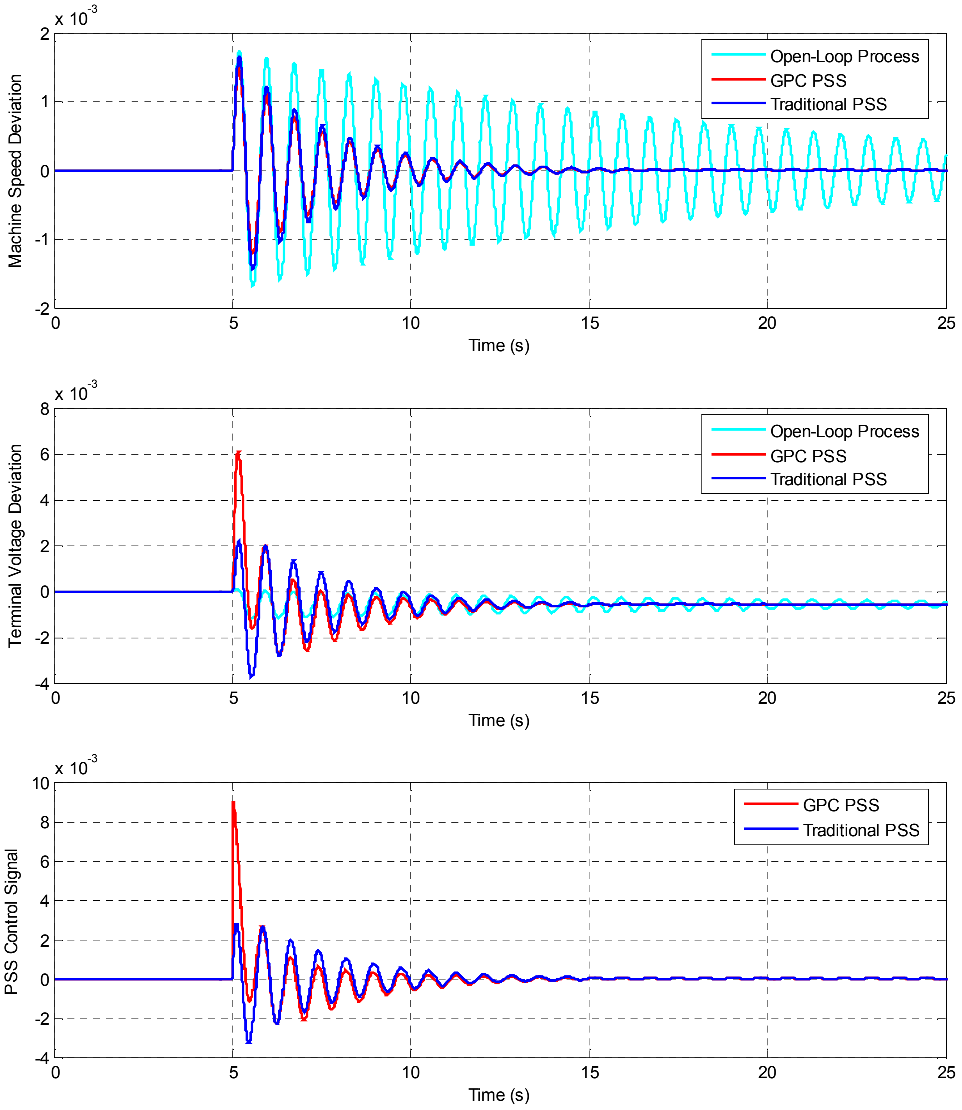

Figure 3-16: Comparison of the time responses from the closed-loop systems employing Traditional PSS #1 and GPC PSS #1. The responses of the open-loop system are added for the sake of reference.

Examining first the machine speed deviation response it becomes clear that the two controllers provide almost identical speed regulation. Both provided markedly better performance than the open loop process. The control signal and terminal voltage deviation for the GPC PSS are fairly similar to those of the traditional PSS except that they have decaying DC components. Both control signals settle to zero in the steady state and the terminal voltage deviations settles to the same steady state value. The small offset is due to a deficiency in the AVR design.

Traditional PSS #2 vs GPC PSS #2

Traditional PSS #2 and GPC PSS #2, being designed to resolve *Control Problem #2*, each achieved a minimum damping of 40% for all of their closed-loop modes. Achieving this target for the GPC controller required some minor deviation from the guidelines for control parameter selection outlined in section 3.3.2. Table 3-4 compares the dominant closed-loop mode from the two closed-loop systems.

	Eigenvalues	**|Eigenvalues|**	**Damping Ratio (%)**	**Frequency (Hz)**
Traditional PSS #2	0.97682 ± 0.049644i	0.97808	40.0000	0.8818
GPC PSS #2	0.95928 ± 0.083325i	0.96289	40.0000	1.5046

Table 3-4: Dominant modes and their properties for closed-looped processes employing Traditional PSS #2 and GPC PSS #2 designs.

Unlike before, the dominant closed-loop modes are quite different. Although they possess equal damping, their oscillation frequencies differ appreciably from each other and from that of the open-loop system. The change in dominant mode frequency is less for the GPC PSS than for the traditional PSS. Based on these observed differences in the dominant modes of the closed-loop systems their time domain responses are expected to vary visibly. Figure 3-17 provides this comparison. Again the disturbance takes the form of a 0.1 pu step increase in the machine's mechanical torque input and is applied at time t = 5s.

The initial swing in the machine speed was larger for the closed-loop process with the traditional PSS than it was for the system with the GPC PSS. The frequency of the first swing when the GPC PSS was employed, consistent with the results of table 3-4, was relatively fast when compared to that which is observed when the traditional PSS is employed. Expectedly, the better initial response of the GPC controller entailed increased control activity. Despite its superior initial performance, however, the GPC

PSS is active for a longer duration of time than the traditional PSS and takes longer to reach a new steady state. This lagging performance is believed to be attributable to a relatively slow decaying mode linked to the GPC controller's integrator.

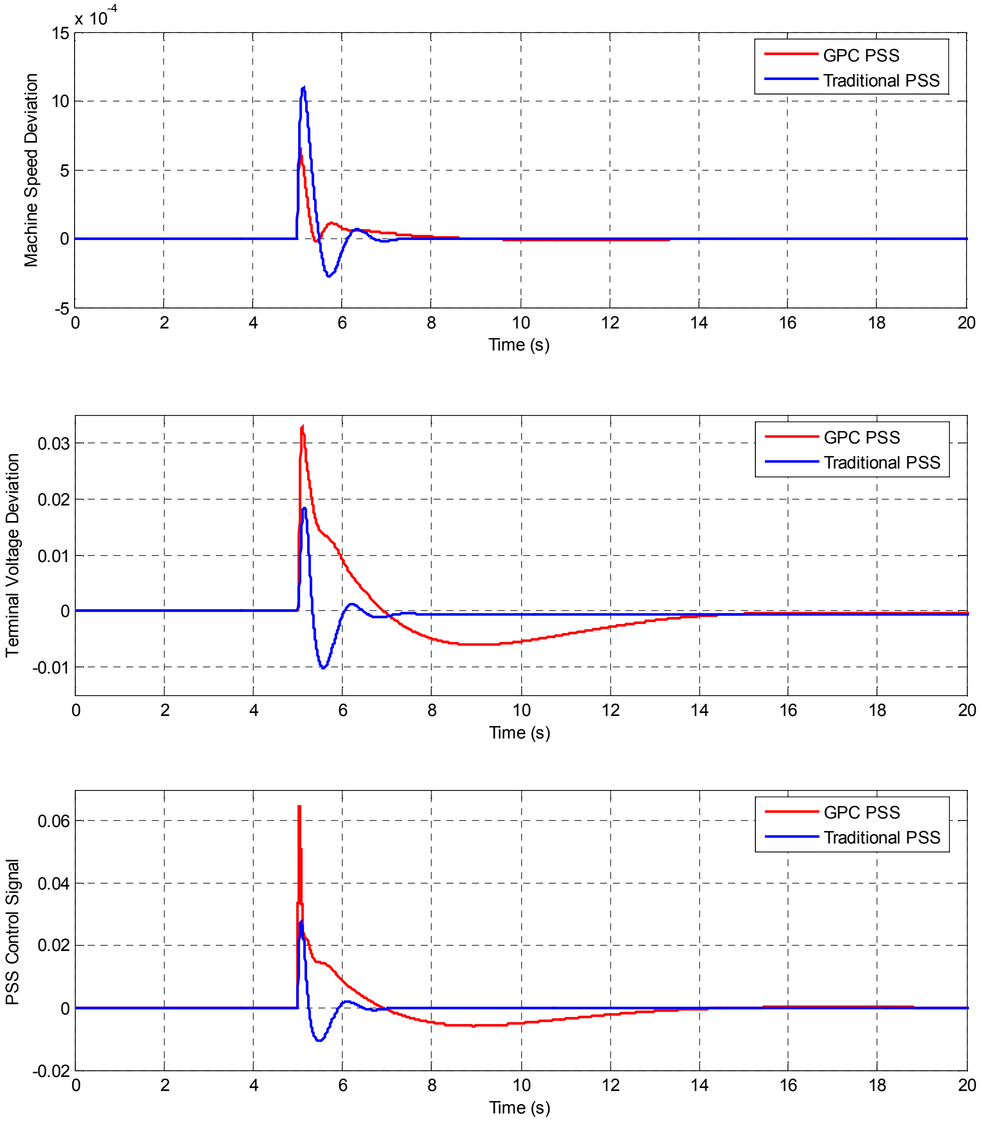

Figure 3-17: Comparison of the time responses from the closed-loop systems employing Traditional PSS #2 and GPC PSS #2.

Traditional PSS #3 vs GPC PSS #3

In this third comparison the capabilities of the controllers were pushed to their limits. Table 3-5 lists the dominant modes from the closed-loop systems that result from the application of these optimized controllers.

	Eigenvalues	|Eigenvalues|	Damping Ratio (%)	Frequency (Hz)
Traditional PSS #3	0.88112 ± 0.18875i 0.98062 ± 0.037445i	0.90111 0.98134	44.2520 44.2600	3.7452 0.6774
GPC PSS #3	0.99471 ± 0.0039839i 0.88198 ± 0.081246i	0.99472 0.88572	79.7360 79.7340	0.1056 2.4224

Table 3-5: Dominant modes and their properties for closed-looped processes employing Traditional PSS #3 and GPC PSS #3 designs

The first point to notice about the results of table 3-5 is the significant difference in the ceiling for the achievable damping. For this particular case the performance ceiling was markedly higher for the GPC PSS than it was for the Traditional PSS. Each closed-loop system contains two pole pairs with approximately the same level of damping but all at very different oscillation frequencies from each other and from that of the open-loop process. Figure 3-18 shows several signals from the time domain simulations. Like the previous two cases, the disturbance is a 0.1 pu step increase in the mechanical torque input to the machine and is applied at time t = 5s.

The damping advantage enjoyed by the GPC PSS relative to the traditional PSS was not reflected in its time responses. The two controllers performed equally well in terms of speed regulation but the control signal of the traditional PSS is favorable because it settles to zero faster, leading to faster voltage regulation. As with the previous comparison, the lossy integrator and the slow decaying closed-loop mode it generates is believed to be the reason why the expected performance advantages where not realized.

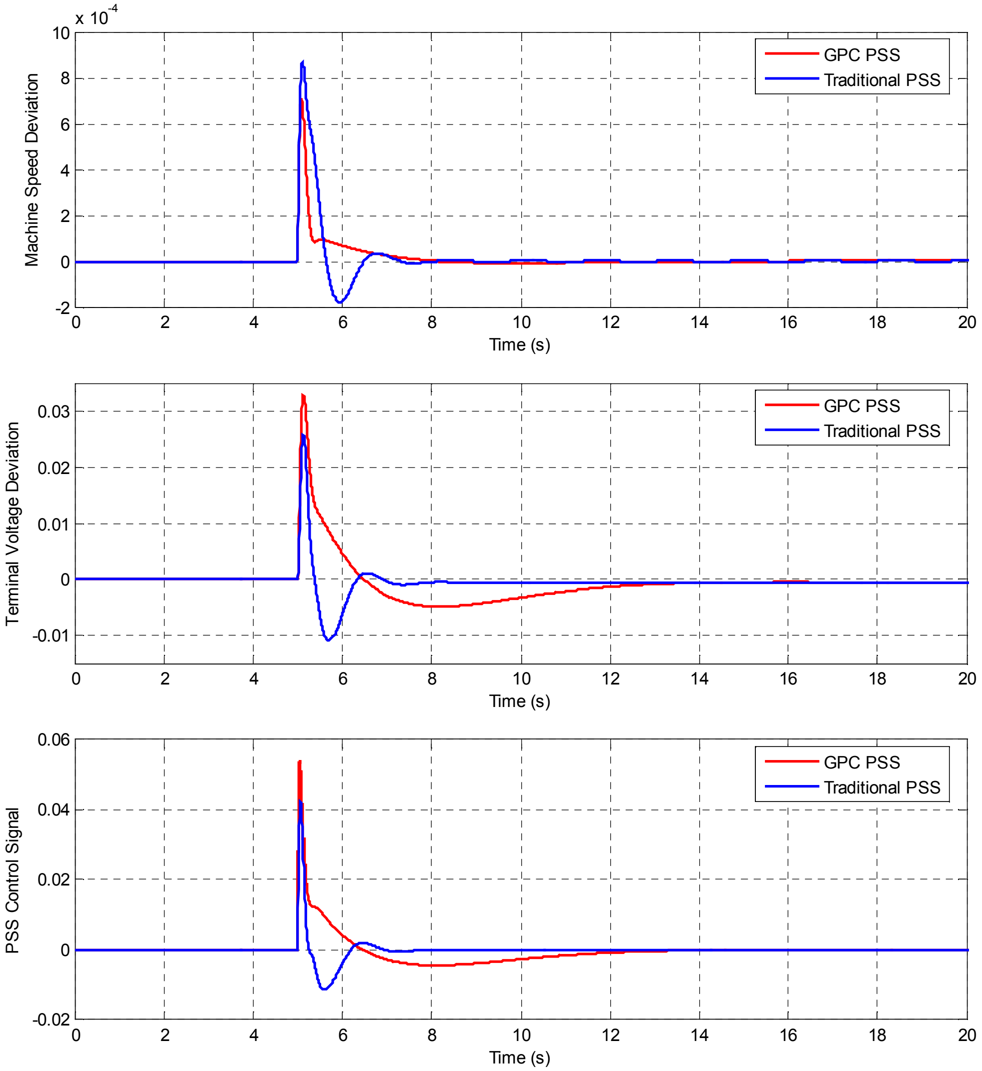

Figure 3-18: Comparison of the time responses from the closed-loop systems employing Traditional PSS #3 and GPC PSS #3.

3.5.2 Off-Nominal Operating Point Performance

One of the reasons why the GPC algorithm was initially selected was its supposed robustness in dealing with processes that either had unknown order or that were changing with time [3]. An attempt to gauge how the robustness of a GPC PSS compares to that of a traditionally designed PSS follows. Each of the controller designs examined in the previous subsection was applied to a series of linearized systems, each representing an off-nominal operating point. The various off-nominal operating points were established by changing one of a number of different system parameters while keeping the others at their nominal values.

Traditional PSS #1 vs GPC PSS #1

Figure 3-19 shows how the dominant closed-loop damping provided by Traditional PSS #1 and by GPC PSS #1 compare as different system operating point changes take place. The two controllers provide closed-loop performance that is very comparable over all examined operating points. The GPC PSS, though, does appear to be slightly more robust, remaining stable for a wider range of operating points than the traditional PSS. Whenever off-nominal operation caused the closed-loop damping to fall below the design target, the GPC PSS controlled process consistently provided better performance than the system employing a traditional PSS.

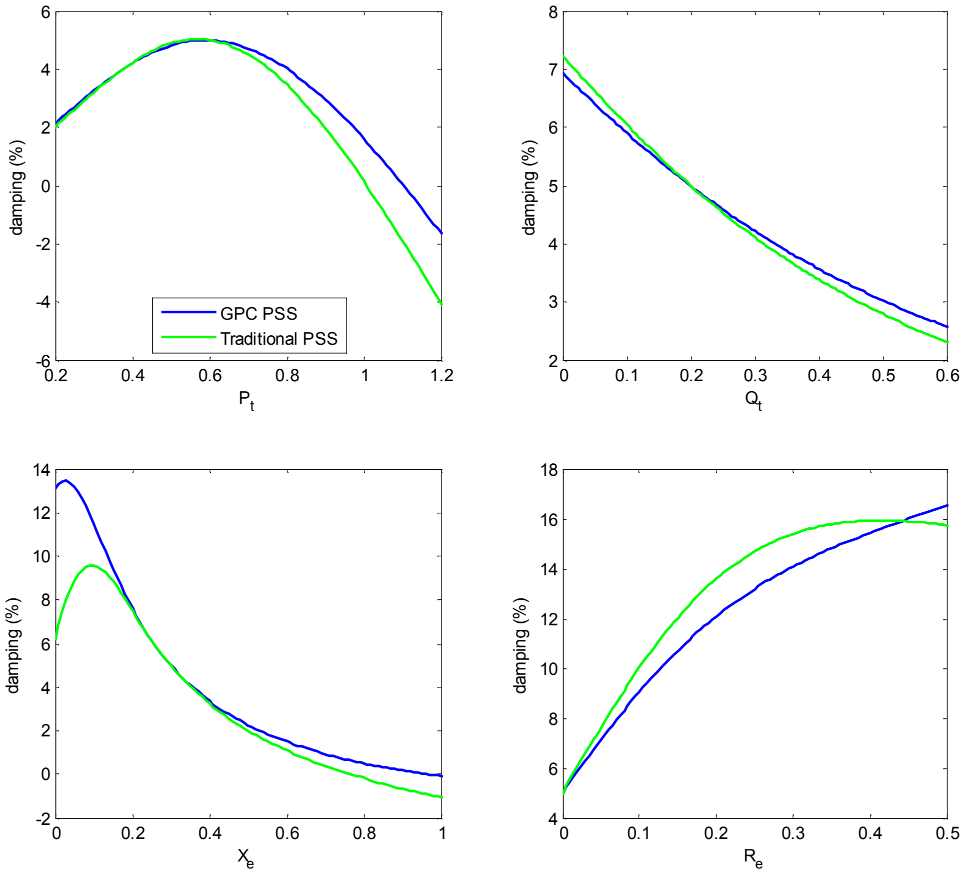

Figure 3-19: Comparison of damping provided by GPC PSS #1 and Traditional PSS #1 during off-nominal closed-loop operation.

Traditional PSS #2 vs GPC PSS #2

Figure 3-20 compares the different damping trajectories that result from applying Traditional PSS #2 and GPC PSS #2 to various off-nominal operating points. The trajectories of the two controllers differ more than those of Traditional PSS#1 and GPC PSS #1. Once again it holds true that when off-nominal operation causes damping performance to dip below the design target, the GPC PSS usually provided slightly better damping than the traditional design.

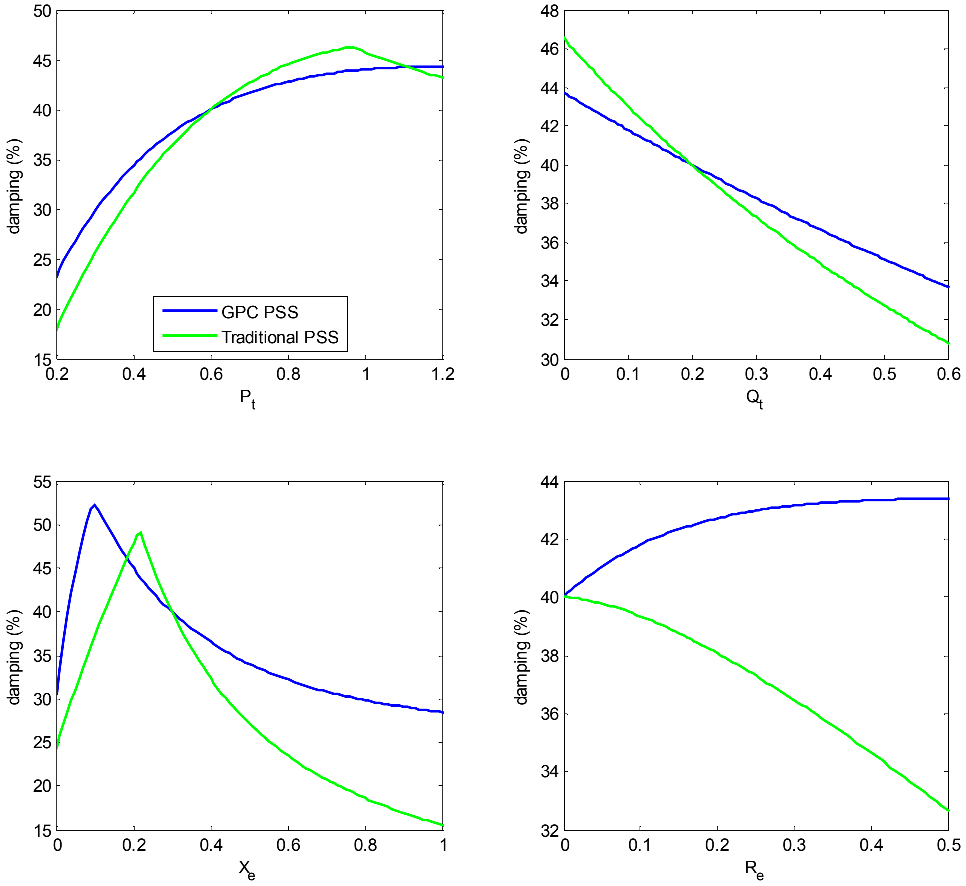

Figure 3-20: Comparison of damping provided by GPC PSS #2 and Traditional PSS #2 during off-nominal closed-loop operation.

Traditional PSS #3 vs GPC PSS #3

The final comparison is done between controllers that were designed to maximize the achieved damping. It was shown earlier that during nominal operation, the GPC possessed a higher performance ceiling. Figure 3-21 shows that this higher level of performance wasn't achieved at the expense of robustness since the GPC PSS provided superior damping than the traditional PSS over all the examined operating points.

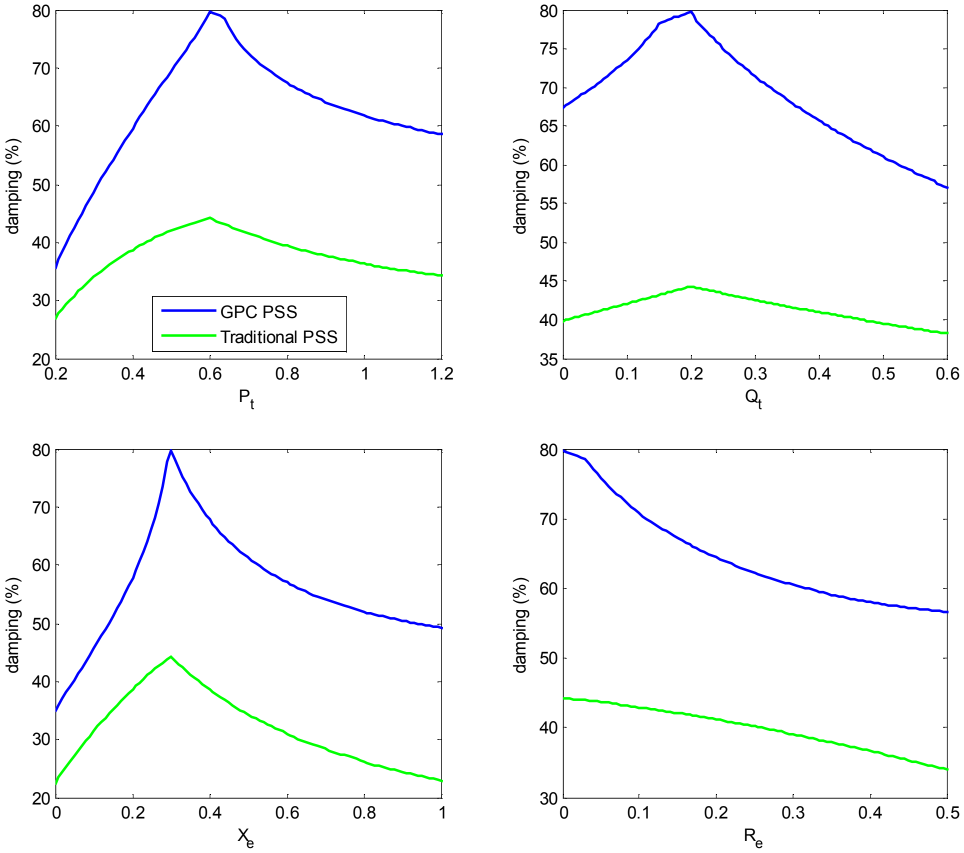

Figure 3-21: Comparison of damping provided by GPC PSS #3 and Traditional PSS #3 during off-nominal closed-loop operation.

3.6 Summary

This chapter dealt with the tasks of designing and evaluating a PSS based on the GPC control algorithm. The primary interest was to evaluate the suitability of the algorithm for the assigned control task. As such, only analytically calculated models of the process were used in this chapter. The chapter began with a description of the process by which a basic GPC PSS could be constructed and detailed a few issues of that design which needed to be resolved. These issues included excessive controller complexity, controller parameter selection and controller offset elimination.

The results of section 3.2 illustrated that in was in fact possible to synthesize an effective GPC controller based on a simplified model of the process. This simplified model was constructed using a partial fraction expansion of the process in question. Specifically, only those terms associated with the dominant poles of the system were used to synthesize the controller. Use of the reduced-order model did, however, entail some drawbacks. It complicated selection of control parameters and degraded slightly the robustness of the controller to off-nominal operating conditions. Still, these things could be overcome and were fairly minor costs when compared to the benefit of being able to use a simplified model in synthesizing the PSS.

Section 3.3 revealed some interesting trends. It was discovered that there is a wide swathe of control parameter combinations that can lead to good performance. Although selection of appropriate control parameters is closely tied to the sampling frequency of the process model, generalizations regarding parameter selection are possible. The investigation conducted here seems to suggest that any modification of the control parameters leading to an increased emphasis of the process model in the derivation of the control increment should not be done in excess. This includes increasing the output horizon, N, decreasing the control horizon, N_u, and decreasing the control penalty, α. The consequence of doing so is that effective control becomes too strong a function of having an accurate process model. When this happens difficulties can arise in any situation where the process being controlled doesn’t correspond to the model used to derive the control signal. This condition may arise during the design phase due to use of the reduced-order model or it may arise during off-nominal operation. Although the process model shouldn’t be overly relied on, the output horizon must be chosen large enough; the control horizon small enough; and the control penalty small enough so that effective control is possible. A balance must be achieved. It is not terribly useful to quantify the qualitative descriptions 'small enough' and 'large enough' because they are dependent on the sampling frequency. Appropriate values need to be determined empirically by trial and error.

Several schemes aimed at removing the control signal offset inherent to the original GPC algorithm were studied. In the end introducing a slight loss to the integrator, essentially giving it a finite memory, was deemed the most appropriate course of action. The lossy integrator showed good performance in the time domain and provided unanticipated stability benefits during off-nominal operating conditions.

The final section of the chapter was a comparison between the proposed GPC PSS and a traditional PSS. Three different control problems were laid out and addressed by designing three GPC controllers and three traditional controllers. Those controllers designed for the same control problem where then compared based on both their nominal and off-nominal operating point performance. Of the three formulated control problems, the latter two had impractically stringent performance specifications. They were formulated mainly to test the limits of performance for each of the control approaches. To that end they succeeded; when pushed the differences in control approach for the GPC PSS and the traditional PSS became apparent. Still, the most pertinent observations are those made when comparing traditional PSS #1 and GPC PSS#1 both of which were designed to meet more practical design targets. These two controllers provided almost identical performance at the nominal operating point. During off-nominal operation the GPC PSS exhibited better performance than the traditional PSS anytime performance dipped below the design target. Based on the results of this chapter it can be stated with a fair amount of confidence that the GPC PSS, for the small systems investigated here, provides comparable performance and would serve as a suitable alternative to a traditional PSS.

4 Adaptive Controller

The previous two chapters of this work examined the problem of system identification and that of controller synthesis as two distinct and separate issues. In this chapter the complicated procedure of unifying the two is started. What follows is only a first step. Discussion is broken down along two lines. First the suitability of different identified models for controller synthesis is examined. Secondly, the challenges of tuning an adaptive GPC PSS are discussed.

4.1 Viability of an estimated-model based GPC PSS

In this first section an attempt is made to assess how the use of an estimated model impacts the performance of a GPC PSS synthesized using it. There is particular interest in how signal quality and the model order assumption affect the effectiveness of the designed controller. The controllers that are synthesized using an estimated model of a process will be referred to as *estimated-model based GPCs*. Likewise, controllers synthesized using an analytically calculated model of a process will be referred to as *analytic-model based GPCs*.

4.1.1 Correct Model Order: $N_a = 4$, $N_b = 4$ and $N_c = 0$

Each curve of figure 4-1 shows the trajectory of the damping for the dominant closed-loop mode that results from the application of a different GPC controller. The first curve makes use of the analytic-model based GPC and is included for the sake of reference. The subsequent four curves use the estimated-model based GPCs. The SNRs of the signals used to construct the process models are indicated in the legend.

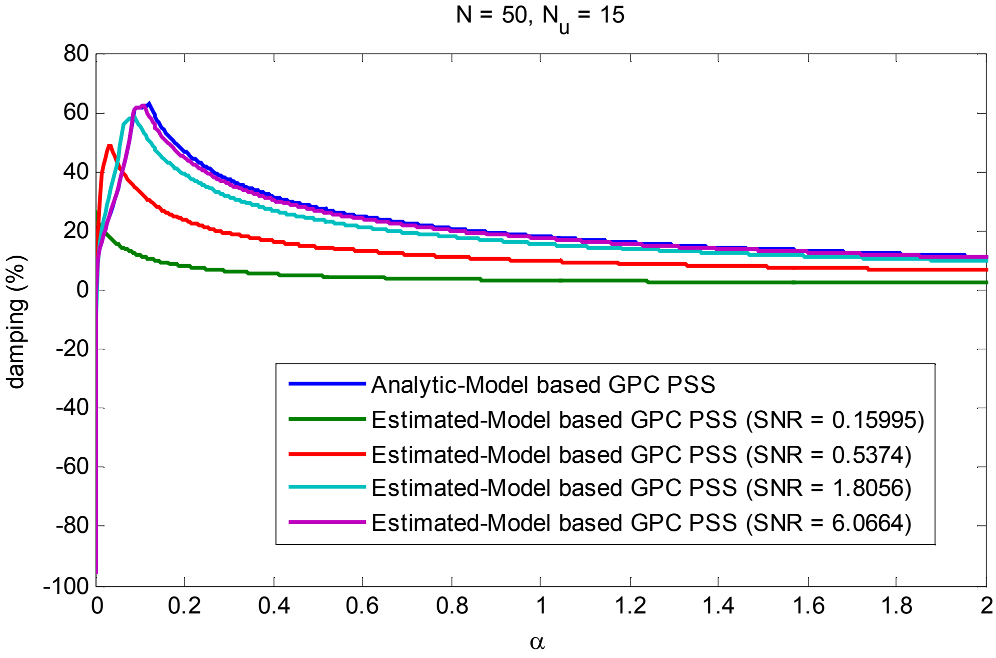

Figure 4-1: Comparison of the damping trajectories for the dominant closed-loop mode that results from application of different estimated-model based GPCs.

The damping trajectories of figure 4-1 suggest that when a process model is constructed using signals having an SNR exceeding roughly 5, then the analytic and estimated models can be used interchangeably to synthesize the controller without sacrificing much performance at the nominal operating point. The damping trajectories for the estimated-model based GPCs converge to that of the analytic-model based GPC as SNR of the signal used to construct the estimated models grows. The reader should also note that although not as effective, relatively good damping can still be achieved from models derived from signals having lower SNRs. Achievement of that damping, however, would require selection of a different set of control parameters than those appropriate for use with the analytic-model based GPC.

Figure 4-1 above was generated using values for the output and control horizons of 50 and 15 respectively. In general the SNR at which the performance of the estimated-model based GPC converged to that of the analytic-model based GPC became increasingly smaller as measures were taken to reduce the influence of the model in the derivation of the control signal. Theoretically such measures include reduction of the output horizon, increasing the control horizon and increasing the control penalty. In practice, manipulation of the output horizon seems most effective at controlling the GPC algorithm's sensitivity to model error.

Figure 4-2 illustrates how the closed-loop performance of some estimated-model based GPC PSSs compare to that of an analytic-model based GPC PSS. The plots show the time responses of the machine speed deviation and GPC control signal after the system is subjected to a 0.1 pu step increase in ΔT_m at time t = 5s. The SNRs of the signals used to construct the models employed for controller synthesis are indicated. The same control parameters are applied to all controllers and their values are chosen so as to ensure 5% damping for the closed-loop process with an analytic-model based GPC PSS.

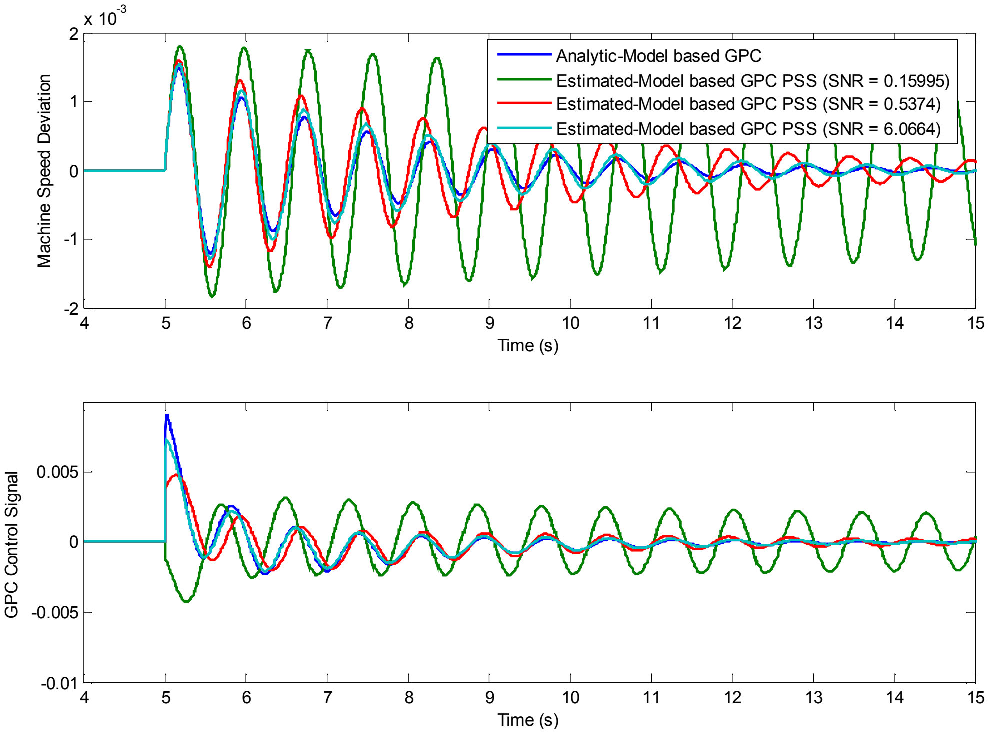

Figure 4-2: Comparison of machine speed deviation and GPC control signals from an analytic-model based GPC PSS and three estimated-model based GPC PSSs. The SNR of the signal used to construct the model for each controller is indicated.

Generally the responses of the systems using the estimated-model based GPCs converged to those of the system using the analytic model based controller as the quality of the estimated model increased. This increase in model quality is synonymous with an increase in the SNR of the signal used to derive that model. Based on the time-domain simulations ran, an estimated-model based GPC PSS synthesized using a model constructed from a signal with a modest SNR of 6.0664 provides performance that is quite similar to that provided by the analytic-model based GPC PSS.

Figure 4-3 shows the time responses of the models used to synthesize the GPC controllers of figure 4-2 when they are subjected to an identical disturbance. The signals of both plots are identical; the second is a magnification of the first. The disturbance takes the form of a 1 pu step increase in the model's voltage reference input. The response of the analytically calculated model is also shown.

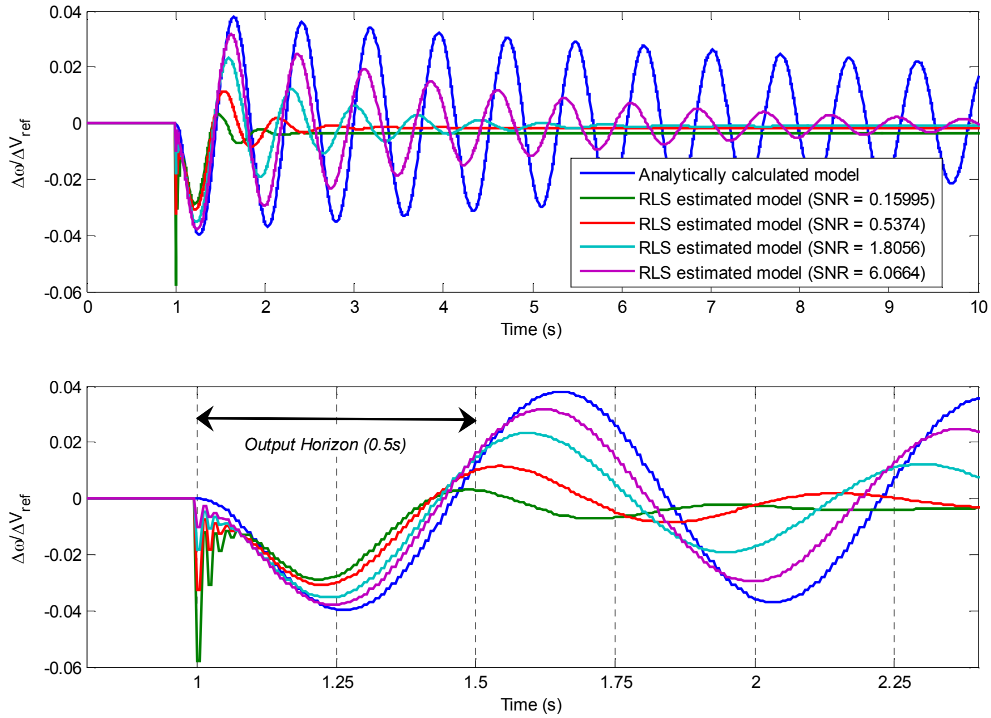

Figure 4-3: Plots comparing time response of different estimated process models to the same disturbance.

Although none of the estimated process models appear especially faithful to the actual process model, they were still useful in the sense that they could be used to design an effective controller. A close examination of the magnified plot of figure 4-3 reveals that over the 0.5s (*N x* T_s) output horizon the estimated models aren't as poor as they appear when they're evaluated over a longer 10s interval. As surmised in Chapter 2, it seems as though a model only needs to be capable of accurately modeling a process over a short interval of time in order to be useful in synthesizing a GPC PSS for it. The length of that interval appears to be dictated by the sampling frequency and the output horizon. It isn't difficult based on the magnified plot to see how using an output horizon of N = 100 (1s) or N = 150 (1.5s) might be problematic.

Table 4-1 list the error in the real and imaginary components of the estimated dominant open-loop mode as well as the resulting error in that mode's damping and frequency. The striking thing about this table is the amount of error in the estimated damping. These results support the theory that accurate knowledge of the dominant mode's damping is not a requisite for the synthesis of an effective controller. The same results also support the theory that reasonably accurate knowledge of the dominant-mode's oscillation frequency is required for the synthesis of an effective controller.

SNR	Eigenvalue Error (%) Real Component	Eigenvalue Error (%) Imag Component	Damping Error (%)	Frequency Error (%)
0.15995	0.039279	0.565780	28.941	0.67978
0.5374	0.022909	0.242100	22.996	0.29705
1.8056	0.008497	0.078646	10.257	0.09248
6.0664	0.002671	0.023477	3.442	0.02680

Table 4-1: Percentage error in the estimated dominant mode and its properties as a function of the SNR of the signal used to construct the associated process model.

Next, a short examination of how the GPC PSS's robustness is affected by the use of estimated-models in their synthesis. Figure 4-4 shows how the dominant closed-loop mode's damping varies with operating point for different estimated model based GPCs. The analytic-model based GPC is also shown in these plots for comparison sake. The trajectories from the estimated-model based GPC controllers appear to converge to those for the analytic-model based GPC as the SNR of the signals used to construct their models is increased. The general shape of the damping trajectories for a specific type of off-nominal operation assumes the same general form regardless of the particular estimated-model based GPC being applied. Performance appears to suffer over all operating points as the quality of the model used to synthesize the controller is reduced.

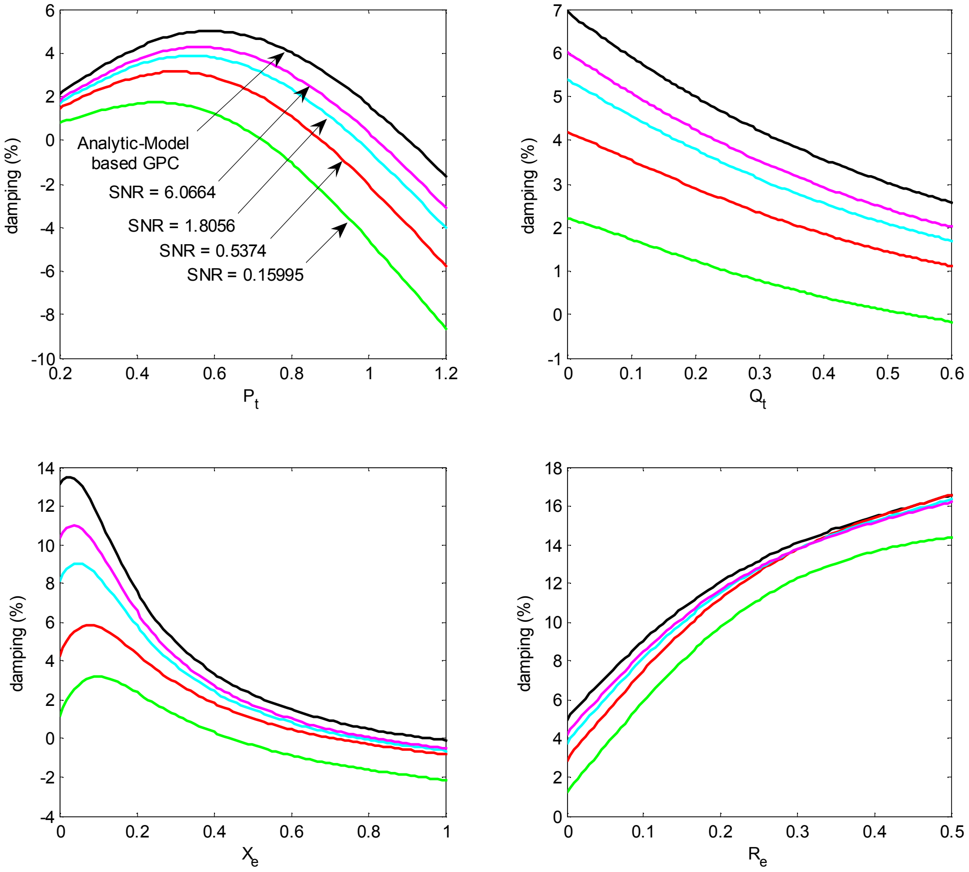

Figure 4-4: Off-nominal operation closed-loop damping provided by several estimated-model based GPC PSSs.

4.1.2 Under-modeling and Over-modeling

So long as the model order assumption was set greater than or equal to the number of dominant poles, similar results to those of section 4.1.1 were obtained. There was, however, some slight variation in the SNR at which the trajectory of the dominant closed-loop mode resulting from application of the estimated-model based controllers converged to that resulting from application of the analytic-model based controller. Nevertheless the difference wasn't great and does not warrant further discussion.

4.2 Control Parameter Selection

In the previous section it was established that given a model derived from signals of sufficient quality, it was possible to achieve closed-loop performance comparable to that which could be achieved using an analytic process model. Potential performance, however, is something different from realized performance. It is the appropriate selection of control parameters that dictate what that realized performance will be. In this section two approaches to control parameter tuning that may be adopted are discussed. The first option is to select the control parameters once and make them fixed. The other option would be to implement a self-tuning scheme where the control parameters are adjusted as the identified model changes.

4.2.1 Fixed Control Parameters:

Using fixed controller parameters is more appealing than a self-tuning scheme because of its simplicity. During the design phase suitable values for the control parameters could be found using an analytic process model valid at the nominal system operating point. These parameters would then remain static regardless of the model that has been identified. The adaptive PSS would then be only adaptive in regards to the model used to derive the control signal.

The suitability of this approach requires that, for the selected combination of control parameters, performance of the closed-loop process varies little with operating point changes. When this is the case, the designer can be reasonably satisfied that good performance will be maintained during off-nominal operation. Figure 4-5 illustrates how the damping trajectories of the dominant closed-loop mode can vary with increasing machine output power. The curves of this plot are generated using the analytical process model.

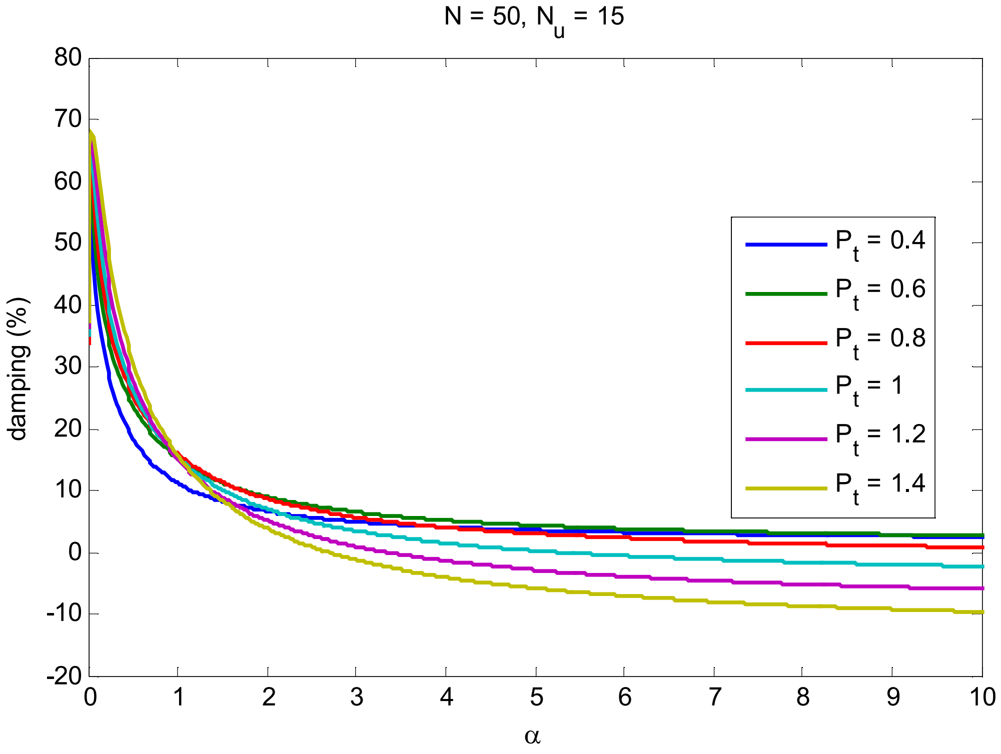

Figure 4-5: Closed-loop damping trajectories for off-nominal operation.

Using similar plots, the degree of change that occurs in the damping performance of the closed-loop system when the operating point changes in the specified manner can be gauged for any specified combination of control parameters.

4.2.2 Self-tuning Control Parameters

An alternative to the fixed controller parameter scheme described above is to make the GPC PSS self-tuning. Embarking on this path would be appropriate in cases where the desired performance during off-nominal operation could not be maintained using a fixed set of control parameters. Unfortunately, implementation of a self-tuning capability would significantly complicate both the design and analysis of an adaptive PSS. Throughout this work choice of appropriate control parameters has been based on the closed-loop performance that results for those choices. More specifically damping of the closed-loop's dominant mode is used a measure of performance. Up until this moment this performance metric was a reliable one because the closed-loop modes and their damping could be accurately calculated since an analytic process model was being used. The luxury of an analytic model is not afforded in the case of a self-tuning controller. The identified process model will not be completely accurate and thus any closed-loop modes calculated from it carry with them some uncertainty. Making decisions based on these incorrect calculations may lead to a poor choice of parameters. Figure 4-6 shows three different configurations used to calculate the closed-loop pole trajectories that are in turn used to determine the best choice of control parameters.

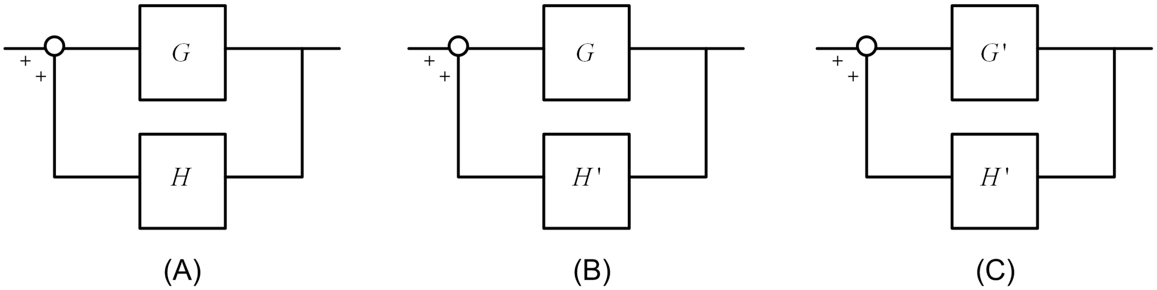

Figure 4-6: Diagram illustrating three principally different ways of calculating closed-loop poles.

Here, G is the actual process model, G' is the RLS estimated process model, H is a GPC controller designed using G and H' is a GPC controller designed using G'. In Chapter 3 configuration A was relied on exclusively to calculate closed-loop properties. In the previous section of this chapter configuration B was used to calculate the closed-loop properties for the estimated-model based GPCs. In a practical situation only configuration C is available for the calculation of closed-loop properties. To illustrate the difficulties presented by not being able to reliably calculate the closed-loop modes consider the plots of figure 4-7. For the different estimated-model based GPCs examined in section 4.1.1, the trajectories of the dominant closed-loop mode’s damping as calculated using configurations B and C are compared. Expectedly, the two trajectories converged more and more as the quality of the estimated-model increased. These plots reveal that even though an estimated model may be sufficient to synthesize an effective controller, it may not always be sufficiently accurate to assess its performance.

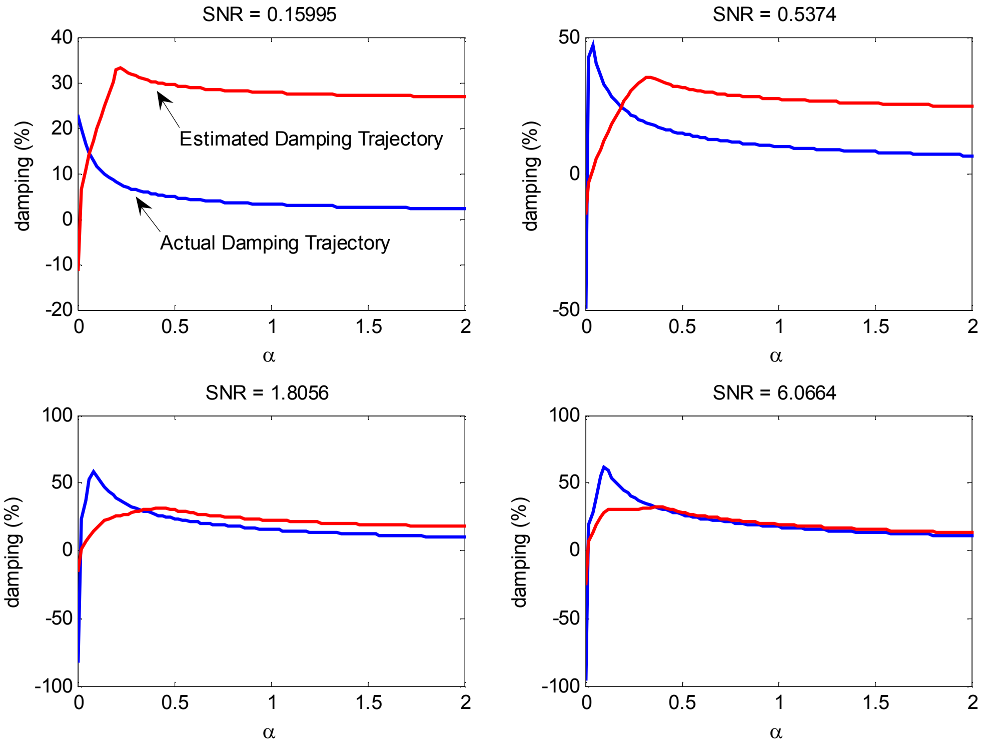

Figure 4-7: Comparison of estimated and actual damping trajectories for different estimated-model based GPC PSSs. (N =50, N_u =15)

4.3 Summary

This chapter has presented a preliminary examination of the performance that could be expected from an adaptive GPC PSS. Parameter estimates yielded by the RLS identification algorithm were used to synthesize GPC controllers.

The results of this chapter's first section have affirmed two hypotheses that were advanced in Chapter 2. First, that a process model only needs to be accurate over a short interval of time in order to synthesize an effective controller. Second, that the model must accurately characterize the system's dominant oscillation frequency if it is to be useful in synthesizing an effective controller. So long as the model order was set at minimum to the number of dominant eigenvalues then model order assumption did not seem to be a critical design choice. Based on the cumulative results presented thus far in this work, a signal of sufficient quality appears to be one having a pre-filter SNR between

5 and 10. Such signals should be sufficient to track system frequency changes and should be suitable to synthesize an effective controller. Using models constructed from signals of this quality caused the performance of the estimated-model based GPCs to lag slightly that of the analytic-model based GPC in terms of both nominal and off-nominal performance. Minor performance drops, however, are of no consequence given that the estimated model is used to facilitate the adaptation that allows the controller to continually operate near the operating point for which it was designed.

In the second section two different approaches to tuning the adaptive controller were discussed. This discussion is important since proper tuning is the key to realizing the performance discussed in the first section of this chapter. The first tuning scheme used fixed control parameters and the second used self-tuning parameters. Some of the perceived merits and deficiencies of each approach were mentioned. In summary, using fixed control parameters is preferable because of its simplicity but it may not deliver the required performance during off-nominal operation. In such a situation a self-tuning approach would be needed. The design of such a self-tuning scheme hasn’t been addressed in this work but would undoubtedly add significant complication to the controller and by extension to its design and analysis.

5 Conclusion

The proceeding work had as its original aim the design of an adaptive power system stabilizer. The design problem was broken down and investigated separately as first a system identification problem and then a control problem. A more holistic treatment of these two components then followed.

The results of Chapter 2 allowed for some generalizations about the RLS algorithm. Signal quality was by far the most significant factor influencing identification. Given a sufficiently high signal quality, the RLS algorithm was capable of constructing very good models. The model order assumption, by comparison, was less of a determining factor in the construction of a good model. At a minimum, though, the assumed model order should be set equal to the number of dominant eigenvalues. Also, good model adaptability appeared to be achievable without incurring too great a risk of algorithmic instability.

In Chapter 3 the design of a GPC algorithm based PSS was detailed. A basic design was presented and the most significant flaws of that design were pointed out and addressed. The finalized GPC PSS was compared to a conventional PSS, leading to some promising results. Although the GPC PSS was generally more active than the traditional PSS it usually exhibited better robustness performance and possessed a higher ceiling on its achievable system damping.

The next chapter combined the works of Chapters 2 and 3 by using the RLS estimated models to construct GPC based PSSs. Results were again encouraging. Using models constructed from signals having modest pre-filter SNRs less than 10 appeared sufficient to both differentiate operating point changes and to synthesize an effective controller. Performance of an estimated-model based GPC PSS converged to that of an analytic-model based GPC as the SNR of the signal used to construct the model grew larger. A given model appeared to be suitable for synthesizing a GPC PSS when it could accurately identify the process' dominant frequency of oscillation. This is surmised based on both reasoning and on experimental observation.

The author finds the most promise in the results obtained from Chapter 2 where the GPC algorithm was synthesized from the analytical model of the process. The analytic-model based GPC PSS compared well to a traditional PSS synthesized using the same model and exhibited better performance in certain regards. Conceptually it is quite different than a traditional PSS but is nevertheless designed using an analytically calculated model of the process. Being a fixed parameter controller, an analytic-model based GPC PSS lends itself to analysis within the frameworks used today in industry.

Taken as a whole, the author is only cautiously optimistic about the practical usefulness of the design presented in this work. While the estimated-model based GPC PSS has not outright failed in a way that would dismiss it, it has not been subjected to enough tests to be considered as anything but an academic exercise. Many concerns regarding how well the RLS algorithm can function in practice persist. Practical systems are nonlinear, unlike the models identified here. Despite the assumptions made here, noise sources are not always separable from the signals of interest. A practical system will have hundreds or thousands of modes contributing in varying degrees all the system's responses, not just the handful observed when an OMIB model is used. Issues pertaining to controller tuning still remain. These are only a few of the issues that cast doubt on the design's practical validity. Given these unresolved concerns the author is unwilling to conclude anything except that the designed controller does show some promise. Further work is needed.

6 Future Work

In regards to furthering the work that was completed in this report, there are a number of avenues that require further research. The ability of the RLS algorithm to identify non-linear power network models must be gauged. Also, the algorithm's performance when applied to larger, multi-machine networks must be assessed. A more detailed investigation into the impact that different types of noise and distortion have on quality of identification should be undertaken. Use of the Recursive Extended Least-Squares (RELS) algorithm as an alternative to the RLS algorithm should be considered. The RELS algorithm is an extension of the RLS algorithm that allows for some limited modeling of noise and distortion. This report's investigations of the GPC algorithm, like those of the identification algorithm, need to be repeated using a more complex multi-machine network model. Validation using non-linear models is less important than with the identification algorithm but is nevertheless worth doing. In terms of advancing the work completed here to another level, two things come to mind. First, the development of a self-tuning scheme for the controller and secondly the implementation of true closed-loop testing. The first is self-explanatory and the second would involve simulations where the estimated parameters from the RLS algorithm are used on the fly by the GPC control algorithm to alter an in-service controller. This form of closed-loop testing would allow for the observation of any emerging behavior due to interaction of the identification and control algorithms. Assuming successful results, a final step would then be to perform extensive real-time testing, a means of providing empirical proof of controller's viability [14].

In regards to the broader topic of an adaptive PSS, more work is needed in defining the problem intended for resolution. The expected measurable benefits from employing this type of a controller must be elaborated. The duration and amount of tolerable deliberate system excitation/disturbance for the sake of system identification, if any, needs to be defined. Specifications on required speeds of adaptation must be developed. In sum, better definition of the problem is needed so that design of a controller to address that problem can proceed in an efficient, systematic manner.

Beyond the question of whether an adaptive controller is possible in a practical network and under practical operating conditions is the question of whether it is desirable. Having an adaptive controller in the network changes how its analysis and planning must be carried out. Are the benefits it offers sufficient to justify the added complexities it introduces?

APPENDIX A

Test Systems

The investigations undertaken in this work used two different test systems. Both are ideal one-machine infinite bus (OMIB) topologies whose basic construction is given in figure A-1. The block diagram of figure A-2 shows the block diagram of the voltage regulation control system.

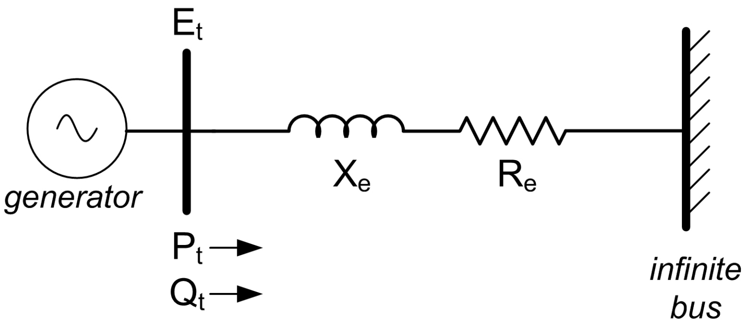

Figure A-1: Basic construction of OMIB test networks

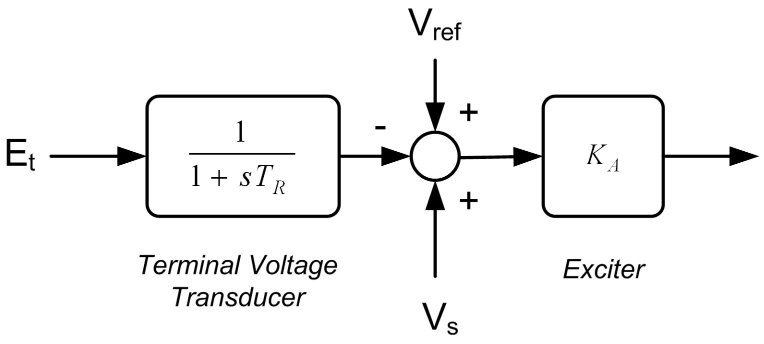

Figure A-2: Voltage Regulator

The test systems will be labeled *Test System 1* and *Test System 2* respectively. The first of these test systems is poorly damped and the second in unstable. Below the nominal operating conditions of the each system is listed. A superficial analysis of each system follows.

Test System 1: *Poorly Damped System*

```
Machine Properties:                                             Saturation:
    Lad   = 1.65      Ra    = 0.003     H     = 3.5                   non-linear              linear
    Laq   = 1.60      Rfd   = 0.0006    Kd    = 0                     Ksd   = 0.8491          Ksd   = 0.434
    Ll    = 0.16      Lfd   = 0.153                                   Ksd   = 0.8491          Ksd   = 0.434

Exciter:              Line Impedance:                           Operating Point:
    Ka    = 200             Xe    = 0.3                               Pt    = 0.6
    Tr    = 0.02            Re    = 0                                 Qt    = 0.2             Et    = 1.00
```

These parameters can be used to construct a linearized state-space model of the process according to a procedure outlined in [1]. The resulting model is given in equation A-1. A discretized version of A-1 is given in A-2. A sampling period of 0.01s is used. Details regarding the discretization process are given in appendix B.

$$\overset{\bullet}{\begin{bmatrix} \Delta\omega(t) \\ \Delta\delta(t) \\ \Delta\Psi_{fd}(t) \\ \Delta v_1(t) \end{bmatrix}} = \begin{bmatrix} 0 & -0.18185 & -0.13861 & 0 \\ 376.99 & 0 & 0 & 0 \\ 0 & -0.2174 & -0.52229 & -27.418 \\ 0 & 0.35972 & 17.266 & -50 \end{bmatrix} \begin{bmatrix} \Delta\omega(t) \\ \Delta\delta(t) \\ \Delta\Psi_{fd}(t) \\ \Delta v_1(t) \end{bmatrix} + \begin{bmatrix} 0.14286 & 0 \\ 0 & 0 \\ 0 & 27.418 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta T_m(t) \\ \Delta V_{ref}(t) \end{bmatrix} \qquad \text{[A-1]}$$

$$\begin{bmatrix} \Delta\omega[n+1] \\ \Delta\delta[n+1] \\ \Delta\Psi_{fd}[n+1] \\ \Delta v_1[n+1] \end{bmatrix} = \begin{bmatrix} 0.99658 & -0.0018147 & -0.0013713 & 0.00016094 \\ 3.7656 & 0.99658 & -0.0025974 & 0.00021086 \\ -0.004621 & -0.0025684 & 0.97477 & -0.21346 \\ 0.005545 & 0.002646 & 0.13442 & 0.58955 \end{bmatrix} \begin{bmatrix} \Delta\omega[n] \\ \Delta\delta[n] \\ \Delta\Psi_{fd}[n] \\ \Delta v_1[n] \end{bmatrix} + \begin{bmatrix} 0.0014269 & -0.0001889 \\ 0.0026913 & -0.00023788 \\ -2.1443\text{e-}006 & 0.27155 \\ 2.7787\text{e-}006 & 0.020059 \end{bmatrix} \begin{bmatrix} \Delta T_m[n] \\ \Delta V_{ref}[n] \end{bmatrix} \qquad \text{[A-2]}$$

Tables A-1 and A-2 summarize the eigen-properties of the continuous-time and the discrete-time state equations of A-1 and A-2.

Eigenvalues	Damping Ratio (%)	Frequency (Hz)
-0.070097 + 8.194j	0.86%	1.3041
-0.070097 - 8.194j	0.86%	1.3041
-37.055	100.00%	0
-13.327	100.00%	0

Table A-1: Eigen-properties of Test System 1 calculated using the continuous-time state-space equation

Eigenvalues	Damping Ratio (%)	Frequency (Hz)
0.99595 + 0.081791j	0.86%	1.3042
0.99595 - 0.081791j	0.86%	1.3042
0.87523	100.00%	0
0.69036	100.00%	0

Table A-2: Eigen-properties of Test System 1 calculated using the discrete-time state-space equation.

Test System 2: *Unstable System*

```
Machine Properties:                                           Saturation:
    Lad   = 1.65      Ra    = 0.003     H     = 3.5                non-linear            linear
    Laq   = 1.60      Rfd   = 0.0006    Kd    = 0                  Ksd   = 0.8491        Ksd   = 0.434
    Ll    = 0.16      Lfd   = 0.153                                Ksd   = 0.8491        Ksd   = 0.434

Exciter:            Line Impedance:                           Operating Point:
    Ka    = 200             Xe    = 0.65                           Pt    = 0.9
    Tr    = 0.02            Re    = 0                              Qt    = 0.3           Et    = 1.00
```

These parameters can be used to construct a linearized state-space model of the process according to a procedure outlined in [1]. The resulting model is given in equation A-3. A discretized version of A-3 is given in A-4. A sampling period of 0.01s is used. Details regarding the discretization process are given in appendix B.

$$\overset{\bullet}{\begin{bmatrix} \Delta\omega \\ \Delta\delta \\ \Delta\Psi_{fd} \\ \Delta v_1 \end{bmatrix}} = \begin{bmatrix} 0 & -0.10918 & -0.12356 & 0 \\ 376.99 & 0 & 0 & 0 \\ 0 & -0.1945 & -0.42432 & -27.418 \\ 0 & -7.3123 & 20.841 & -50 \end{bmatrix} \begin{bmatrix} \Delta\omega \\ \Delta\delta \\ \Delta\Psi_{fd} \\ \Delta v_1 \end{bmatrix} + \begin{bmatrix} 0.14286 & 0 \\ 0 & 0 \\ 0 & 27.418 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta T_m \\ \Delta V_{ref} \end{bmatrix} \tag{A-3}$$

$$\begin{bmatrix} \Delta\omega[n+1] \\ \Delta\delta[n+1] \\ \Delta\Psi_{fd}[n+1] \\ \Delta v_1[n+1] \end{bmatrix} = \begin{bmatrix} 0.99794 & -0.0010935 & -0.0012218 & 0.00014344 \\ 3.7673 & 0.99794 & -0.002315 & 0.00018795 \\ 0.0074785 & 0.0065651 & 0.9716 & -0.21323 \\ -0.11711 & -0.057127 & 0.16213 & 0.58604 \end{bmatrix} \begin{bmatrix} \Delta\omega[n] \\ \Delta\delta[n] \\ \Delta\Psi_{fd}[n] \\ \Delta v_1[n] \end{bmatrix} + \begin{bmatrix} 0.0014269 & -0.00016836 \\ 0.0026913 & -0.00021204 \\ 2.3334\text{e-}006 & 0.27129 \\ -5.8098\text{e-}005 & 0.024205 \end{bmatrix} \begin{bmatrix} \Delta T_m[n] \\ \Delta V_{ref}[n] \end{bmatrix} \tag{A-4}$$

Tables A-3 and A-4 summarize the eigen-properties of the continuous-time and the discrete-time state equations of A-3 and A-4.

Eigenvalues	Damping Ratio (%)	Frequency (Hz)
0.50353 + 7.2331j	-6.94%	1.1512
0.50353 - 7.2331j	-6.94%	1.1512
-31.036	100.00%	0
-20.396	100.00%	0

Table A-3: Eigen-properties of Test System 1 calculated using the continuous-time state-space equation.

Eigenvalues	Damping Ratio (%)	Frequency (Hz)
1.0024 + 0.072633j	-6.94%	1.154
1.0024 - 0.072633j	-6.94%	1.154
0.73319	100.00%	0
0.8155	100.00%	0

Table A-4: Eigen-properties of Test System 1 calculated using the discrete-time state-space equation

Additional Comments:

Consider an operating point change triggered by either an increase or a decrease in the machine output power. For the most part a smooth progression in the eigenvalues will be observed but eventually a threshold will be crossed where they will change dramatically and suddenly. In particular the dominant open-loop mode will change from being complex to being real. Such a change entails significant ramifications for the dynamic behavior of the process since the dominant modes goes from being oscillatory to non-oscillatory or vice-versa. Table A-5 shows how such a change develops for Test System 2. The transition point is highlighted. Regardless of how small the increment in machine power, the transition never appears smooth. Similar transition points can be observed as other system parameters are varied.

Pt				
0.60	0.99895 + 0.066818i	0.99895 - 0.066818i	0.77418 + 0.056478i	0.77418 - 0.056478i
0.70	0.99976 + 0.068762i	0.99976 - 0.068762i	0.77456 + 0.038199i	0.77456 - 0.038199i
0.80	1.0009 + 0.070729i	1.0009 - 0.070729i	0.78824	0.76100
0.90	1.0024 + 0.072633i	1.0024 - 0.072633i	0.81550	0.73319
1.00	1.0042 + 0.074398i	1.0042 - 0.074398i	0.82883	0.71861
1.10	1.07580	0.89547 + 0.034201i	0.89547 - 0.034201i	0.69913
1.20	1.07670	0.90012 + 0.044765i	0.90012 - 0.044765i	0.69060
1.30	1.07750	0.90411 + 0.052627i	0.90411 - 0.052627i	0.68342
1.40	1.07810	0.90758 + 0.059104i	0.90758 - 0.059104i	0.67727

Table A-5: Evolution of open-loop eigenvalues for change power outputs

When the performance of the synthesized controllers was being examined under off-nominal operating conditions the sudden change in the dynamic properties of the underlying process led to their failure. This was true, almost without exception, for both the traditional and GPC controller designs. In practice, however, these types of abrupt changes in the small signal properties of a network are not encountered; they are anomalies of the OMIB model. Accordingly, the scopes of the off-nominal investigations are limited to avoid crossing the thresholds where such behavior occurs. The occasional exception does appear whenever instructive.

APPENDIX B

Calculation Details

State Equation Discretization:

A continuous-time state space model can be converted in a discrete-time shift operator state state-space model using the *step-invariant transformation* [18].

continuous-time state equations $\qquad\qquad$ discrete-time state equations

$$\begin{aligned}\frac{d\mathbf{x}(\mathbf{t})}{dt} &= \mathbf{A}\cdot\mathbf{x}(\mathbf{t})+\mathbf{B}\cdot\mathbf{u}(\mathbf{t}) \\ \mathbf{y}(\mathbf{t}) &= \mathbf{C}\cdot\mathbf{x}(\mathbf{t})+\mathbf{D}\cdot\mathbf{u}(\mathbf{t})\end{aligned} \quad \Longrightarrow \quad \begin{aligned}\mathbf{x}[\mathbf{n+1}] &= \mathbf{\Phi}\cdot\mathbf{x}[\mathbf{n}]+\mathbf{\Gamma}\cdot\mathbf{u}[\mathbf{n}] \\ \mathbf{y}[\mathbf{n}] &= \mathbf{C}\cdot\mathbf{x}[\mathbf{n}]+\mathbf{D}\cdot\mathbf{u}[\mathbf{n}]\end{aligned} \qquad \text{[B-1]}$$

where

$$\mathbf{\Phi} = \mathbf{e}^{\mathbf{A}\cdot T_s} \qquad\qquad T_S = \textit{sampling period}$$

$$\mathbf{\Gamma} = \mathbf{A}^{-1}(\mathbf{\Phi}\,\text{-}\,\mathbf{I})\mathbf{B}$$

Bilinear Transformation:

Often it is necessary to convert a continuous-time domain transfer function to a sample domain transfer function. The *bilinear transformation* [5] was useful for this purpose.

$$s \;\approx\; \frac{2}{T_s}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \qquad \textit{or} \qquad z \approx \frac{1+sT_s/2}{1+sT_s/2} \qquad \text{[B-2]}$$

where T_S = *sampling period*

An instructive example for the application of the bilinear transformation is the conversion of the continuous-time washout filter transfer function given in equation B-3 to the sample domain.

$$H(s) = \frac{sT_w}{1+sT_w} \qquad \text{[B-3]}$$

H(z^{-1}) can be written as given in equation B-4 if the Laplace variable, *s*, is substituted out of B-3 by using the bilinear transformation.

$$H(z^{-1}) = \frac{f_0 + f_1 z^{-1}}{e_0 + e_1 z^{-1}} \qquad or \qquad H(z) = \frac{f_0 z + f_1}{e_0 z + e_1} \qquad [B-4]$$

where

$$f_0 = \frac{1}{(1+K)}, \quad f_1 = -\frac{1}{(1+K)}, \quad e_0 = 1, \quad e_1 = \frac{(K-1)}{(1+K)} \quad and \quad K = \frac{T_s}{2T_w}$$

State-space to transfer function:

Because the control algorithm used in this work requires that the process be cast as a difference equation it is necessary to convert the state-space representation of the process into a transfer function. The transfer function can then readily be converted into a difference equation.

First the state-space equations must be cast into a SISO form. Both y[n] and u[n] of B-1 become scalars representing the system output and system input which are of interest. The columns of Γ from equation B-1 corresponding to all other inputs are removed. Likewise the rows of C and D from equation B-1 that correspond to other system outputs are eliminated. Taking the *z*-transform of the simplified discrete-time state-space equations results in equation B-5. Here, Γ’, C’ and D’ represent the truncated versions of Γ, C and D.

$$\begin{aligned} z\mathbf{X} &= \mathbf{\Phi} \cdot \mathbf{X} + \mathbf{\Gamma}' \cdot U \\ Y &= \mathbf{C}' \cdot \mathbf{X} + \mathbf{D}' \cdot U \end{aligned} \qquad [B-5]$$

Substituting out the state variables leads to the transfer function of equation B-6. This operation can be achieved using the Matlab function *ss2tf()*. [13]

$$\frac{Y}{U} = \left(\mathbf{C}\left(z^{-1} - \mathbf{\Phi}\right)^{-1}\mathbf{\Gamma} + \mathbf{D}\right) \qquad [B-6]$$

Carrying out the operations of the right hand side of equation B-6 leads to a ratio of polynomials that can generally be written as given in B-7. Here, *n* is the order of the process (ie: **Φ** is an *n* x *n* matrix).

$$\frac{Y}{U}=\frac{b_0z^n+b_1z^{n-1}+b_2z^{n-2}+\cdots+b_n}{a_0z^n+a_1z^{n-1}+a_2z^{n-2}+\cdots+a_n} \qquad [B-7]$$

The coefficients b$_0$ will always be zero because of the calculation of (z**I**-**Φ**)$^{-1}$. By definition (z**I**-**Φ**)$^{-1}$T = **adj**(z**I**-**Φ**)/det(z**I**-**Φ**). The elements of the adjoint matrix **adj**(z**I**-**Φ**) will all be polynomials of z each having an order of *n*-1. The polynomial of z resulting from det(z**I**-**Φ**) will be of order *n*. Equation B-7 can thus be rewritten as in equation B-8.

$$\frac{Y}{U}=\frac{b_1z^{n-1}+b_2z^{n-2}+\cdots+b_n}{a_0z^n+a_1z^{n-1}+a_2z^{n-2}+\cdots+a_n} \qquad [B-8]$$

Dividing equation B-8 through by z^n leads to the following:

$$\frac{Y}{U}=\frac{b_1z^{-1}+b_2z^{-2}+\cdots+b_nz^{-n}}{a_0+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}} \qquad [B-9]$$

$$\frac{Y}{Uz^{-1}}=\frac{b_1+b_2z^{-1}+\cdots+b_nz^{-(n-1)}}{a_0+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}} \qquad [B-10]$$

$$\frac{Y}{U'}=\frac{b_0^{'}+b_1^{'}z^{-1}+b_2^{'}z^{-2}+\cdots+b_n^{'}z^{-(n-1)}}{a_0+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}} \qquad [B-11]$$

where

$$Z^{-1}\{Y\}=y[t]$$
$$Z^{-1}\{U\}=u[t]$$
$$Z^{-1}\{U^{'}\}=u[t-1]$$

Equation B-11 is the assumed form of the process for both the RLS identification algorithm and the GPC control algorithm. For this particular form the order of the numerator is one degree less than the denominator.

Closed-loop properties:

In order to evaluate the controllers that were designed a means of calculating the closed-loop properties stemming from application of those controllers was needed. Two different means of doing this were used.

The first method was to simply use a transfer function approach. Figure B-1 illustrates this straightforward method. The only problem is that this approach only gives a limited glimpse into how the process works. Since the process model must first be simplified to a SISO process, much information about it is lost. Still, the poles of the polynomial AM-BN will be accurate.

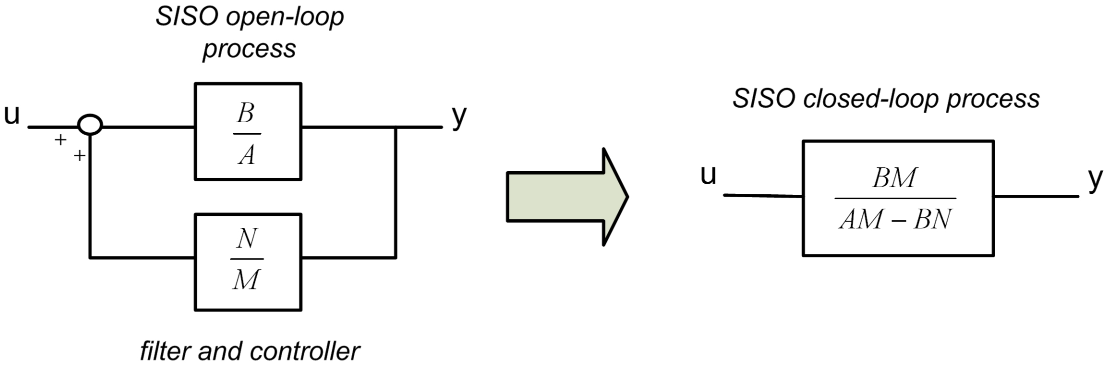

Figure B-1: Calculation of closed-loop poles using transfer functions

An alternative would be to incorporate the designed controller into the state equation. This approach is a little more involved but it is more general and it facilitates the application of a large body of eigenvalue analysis techniques to the closed-loop system. Figure B-2 illustrates the more complete nature of the state space approach to calculating the closed-loop properties. If N(z)/M(z) is written in terms of positive powers of the shift variable, z, then incorporating it into a discrete state-space equation is straightforward. In fact it parallels incorporating a traditional PSS into a continuous-time state-space equation. (see section 12.5 in [1])

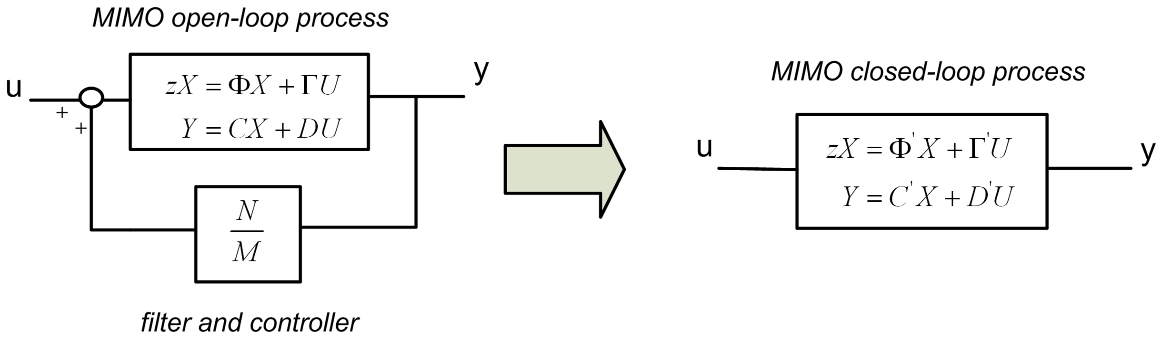

Figure B-2: Calculation of closed-loop poles using state equations

Mode damping and mode frequency

Calculation of the damping ratio, ξ, and the frequency of oscillation, ω, of a mode require the use of different equations depending on whether analysis is being carried out in the continuous-time or discrete-time domains. [8] The relevant equations are summarized below.

For a continuous-time domain pole located at s = u+jv

$$\xi = \frac{-u}{\sqrt{u^2 + v^2}} \cdot 100 \qquad (\,\%\,) \qquad [B\text{-}12]$$

$$\omega = v \qquad \text{(rad/s)} \qquad [B\text{-}13]$$

For a discrete-time domain pole located at z = u+jv = r∠±θ

$$\xi = \frac{-\ln(r)}{\sqrt{\ln^2(r) + \theta^2}} \cdot 100 \qquad (\,\%\,) \qquad [B\text{-}14]$$

$$\omega = \frac{1}{T_S}\sqrt{\ln^2\left(r\right) + \theta^2} \qquad \text{(rad/s)} \qquad [B\text{-}15]$$

where T_s is the sampling period

Signal Qualtity

Signal-to-Noise Ratio (SNR) was one of the metrics used in the work presented here to evaluate the quality of signals used by the identification algorithm. Equation B-16 below summarizes its calculation [12].

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}} \right)^2 \qquad \text{[B-16]}$$

Here, A is the RMS amplitude of a given signal. For a discrete signal given by N values $\{x_1, x_2, ..., x_N\}$ the RMS value is calculated according to equation B-17.

$$\mathrm{A} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} x_i^2} = \sqrt{\frac{x_1^2 + x_1^2 + \cdots + x_N^2}{N}} \qquad \text{[B-17]}$$

APPENDIX C

Traditional PSS Design

In this section the design of a traditional PSS is detailed. The controller is design to meet the specifications of Control Problem #1 listed in section 3.5. That problem is to design a controller for Test System 1 that ensures a minimum of 5% damping for all the closed-loop modes.

The basic design objective of a traditional PSS is to introduce an electric torque that is in phase with any change in speed. The means of introducing this electric torque is through modulation of a generator exciter's reference value, $V_{ref.}$ Unfortunately there is a phase lag introduced to any signal applied to the voltage reference input. The bulk of the work for designing a PSS involves identification and compensation of this phase lag. Figure C-1 shows the basic block diagram a typical PSS. The number of lead-lag block that are needed will depend on the amount of compensation that is required.

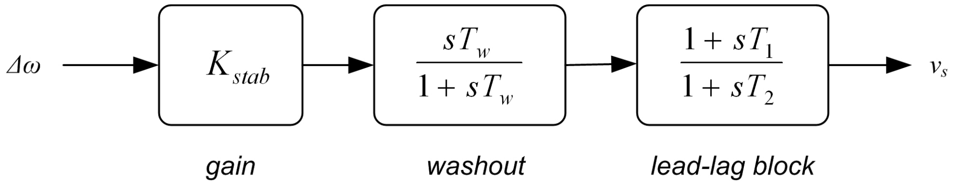

Figure C-1: Block diagram of the components of a traditional PSS.

Washout:

The washout is a high-pass filter that limits the action of controller to slow changes in operating point. It typically has little impact on phase characteristics of the controller and its effect is disregarded in this respect. The washout time constant used here is assigned a value of T_w = 0.5. The lead-lag block compensates for the aforementioned phase lag.

Phase Compensation:

The first task is to identify the process' problem mode. Table A-1 of appendix A is duplicated here for the sake of convenience. It summarizes the open loop poles of Test System 1. The dominant eigenvalues have been highlighted and have damping of only

0.86% and an oscillation frequency of 1.3041Hz. It is for these dominant eigenvalues that the PSS will be designed.

Eigenvalues	Damping Ratio (%)	Frequency (Hz)
-0.070097 + 8.194j	0.86%	1.3041
-0.070097 - 8.194j	0.86%	1.3041
-37.055	100.00%	0
-13.327	100.00%	0

Next, the phase lag between ΔT_e and ΔV_{ref} or equivalently between $\Delta\Psi_{fd}$ and ΔV_{ref} of the systems generator must be found. In doing this $\Delta\omega$ and $\Delta\delta$ remain zero and any feedback path through which either affects $\Delta\Psi_{fd}$ is eliminated. This is achieved by setting the a_{32} and a_{42} elements of the A matrix to zero. Equation C-1 shows how the state matrix is modified.

$$A' = \begin{bmatrix} 0 & -0.18185 & -0.13861 & 0 \\ 376.99 & 0 & 0 & 0 \\ 0 & 0 & -0.52229 & -27.418 \\ 0 & 0 & 17.266 & -50 \end{bmatrix} \qquad [C\text{-}1]$$

The input matrix is also modified by setting ΔT_m to zero since it is of no interest here. The modified input matrix is given in equation C-2.

$$B' = \begin{bmatrix} 0 \\ 0 \\ 27.418 \\ 0 \end{bmatrix} \qquad [C\text{-}2]$$

Using these modified matrices, the desired transfer function is then calculated using the Laplace Transformation as follows. Here the Laplace operator is assigned $s = j\omega$ where ω is equal to the frequency of the dominant mode in rad/s.

$$\dot{x} = A'x + B'u \qquad [C\text{-}3]$$

taking the Laplace Transformation gives

$$sX = A^{'}X + B^{'}U \qquad [C\text{-}4]$$

$$X = \left(s - A'\right)^{-1} B'U \qquad [C\text{-}5]$$

$$\frac{X}{U} = \left(s - A'\right)^{-1} B' \qquad [C\text{-}6]$$

$$\begin{bmatrix} \dfrac{\Delta\omega}{\Delta V_{ref}} & \dfrac{\Delta\delta}{\Delta V_{ref}} & \dfrac{\Delta\Psi_{fd}}{\Delta V_{ref}} & \dfrac{\Delta v_1}{\Delta V_{ref}} \end{bmatrix}^T = \left(s - A'\right)^{-1} B^{'} \qquad [C\text{-}7]$$

The desired transfer function is the third element of the calculated vector from C-7. Setting s = 8.1939j, the frequency of the dominant mode, the transfer function $\Delta\Psi_{fd}/\Delta V_{ref}$ assumes a value of $2.3207\angle{-34.4494}°$. Since $\Delta\Psi_{fd}$ and ΔT_e are approximately proportional to each other −34.4494° is the phase lag introduced to a signal of 8.1939 rad/s as it passes from ΔV_{ref} to $\Delta T_{e.}$ The PSS must be designed to compensate for this phase lag. One lead-lag block is sufficient to provide the required 35° lead. As mentioned, the washout filter has only a small impact on the phase and is therefore neglected.

Having determined the amount of compensation require, the lead-lag time constants can be chosen appropriately using equations C-8 through C-10. Setting θ_m equal to the required amount of phase lead and ω_m to the dominants mode's frequency of oscillation, equation C-8 can be solved for *a*. T_1 and T_2 can then be solved in sequence using C-9 and C-10.

$$\sin(\theta_m) = \frac{1-a}{1+a} \qquad [C\text{-}8]$$

$$T_1 = \frac{1}{\omega_m \sqrt{a}} \qquad [C\text{-}9]$$

$$a = \frac{T_2}{T_1} \qquad [C\text{-}10]$$

Application of these equations to this example leads to $T_1 = 0.2317$ and $T_2 = 0.0643$. Having specified T_w, T_1 and T_2; a root locus plot such as that of figure C-2 can be generated and the gain can be selected to achieve the desired level of closed-loop damping. For the design problem address here, a stabilizer gain of 1.1364 was selected. Table C-1 lists the closed-loop modes that result after application of the PSS.

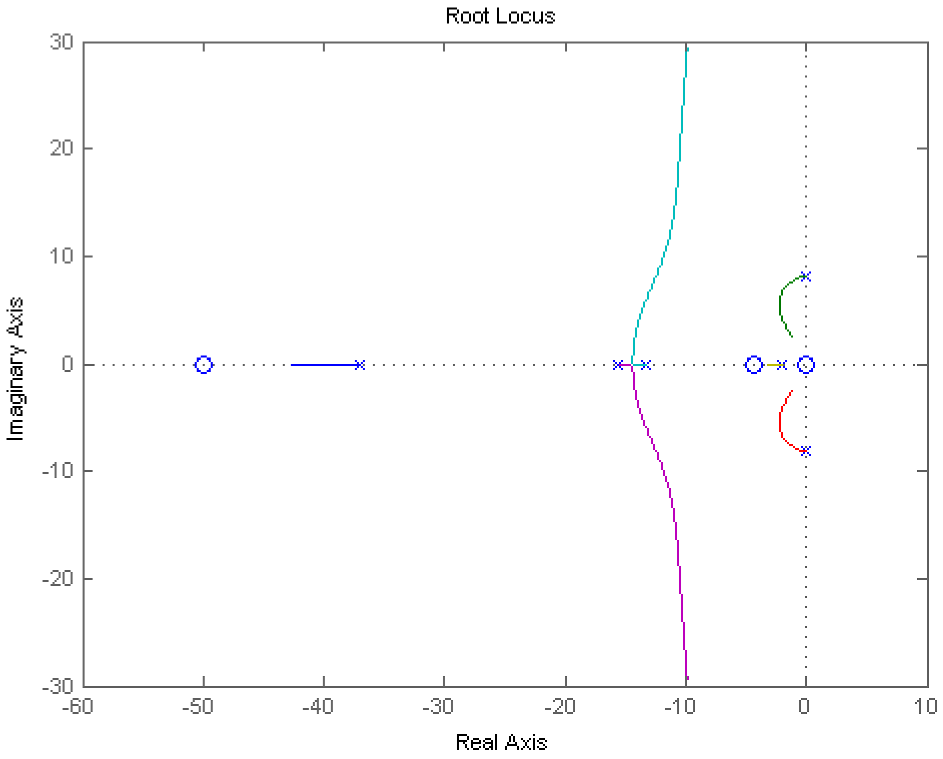

Figure C-2: Root-locus plot of closed-loop system employing a traditional PSS.

Eigenvalues	Damping Ratio (%)	Frequency (Hz)
-37.389	100.0000	0
-0.40519 + 8.0944i	4.9996	1.2883
-0.40519 - 8.0944i	4.9996	1.2883
-13.931 + 3.7524i	96.5590	0.59721
-13.931 - 3.7524i	96.5590	0.59721
-2.0184	100.0000	0

Table C-1: Table of closed-loop modes after application of traditional PSS

APPENDIX D

GPC PSS Design

In this section the design of a GPC PSS is detailed. The controller is designed to meet the specifications of Control Problem #1 that are listed in section 3.5. In sum that problem is to design a controller for Test System 1 that ensures a minimum of 5% damping for all the closed-loop modes.

The first step in the design of a GPC PSS is to acquire a model of the process intended for control. For the reasons enumerated in section 3.2 a reduced-order model has be used in place of the full-order model. Table A-2 of appendix A that summarizes the open-loop modes of Test Systems 1 is repeated here for the sake of convenience. A complex pole pair located at z = 0.995945 ± 0.081791j are identified as the dominant poles having damping of 0.86% and oscillation frequency of 1.3042 Hz. (T_s = 0.01s) The GPC is designed based on these dominant eigenvalues.

Eigenvalues	Damping Ratio (%)	Frequency (Hz)
0.99595 + 0.081791j	0.86%	1.3042
0.99595 - 0.081791j	0.86%	1.3042
0.87523	100.00%	0
0.69036	100.00%	0

The discretized analytic state-space model of equation A-2 was converted into a transfer-function (see appendix B) and then the procedure outlined in section 3.2.1 was used obtain a reduced-order model of that transfer function. The input and output of the constructed transfer function are respectively ΔV_{ref} and $\Delta\omega$. Equation D-1 provides the calculated reduced-order process model.

$$\frac{\Delta\omega}{\Delta V_{ref}} = 10^{-3}\frac{-2.7589+2.6064z^{-1}}{1-1.9919z^{-1}+0.9986z^{-2}} \qquad [D\text{-}1]$$

Equipped with the process model of equation D-1, design of the GPC PSS could proceed. A lossy integrator with an integrator loss factor of 0.995 was employed to eliminate the controller offset and to resolve the problem of marginal instability. Like with a traditional PSS, a washout filter is used to prevent the controller from responding to steady state changes in the operating point. In order to be consistent with the traditional PSS designed in appendix C, the washout time constant was assigned a value of $T_w = 0.5$. It was not, however, considered in synthesizing the controller. Based on the observations and guideline presented in section 3.3 the output horizon was set to 50, the control horizon was set to 15 and the control penalty was set to 6.0958. Figure D-1 shows the plot used to select the control penalty.

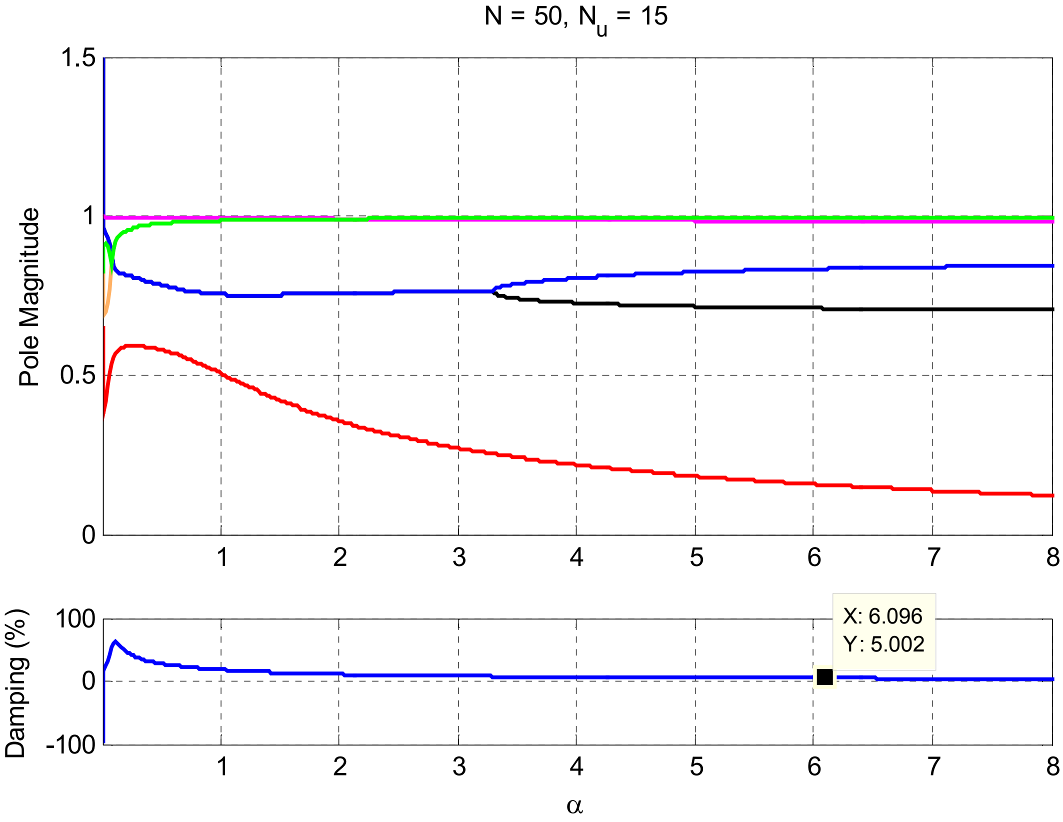

Figure D-1: Plot used to aid in the selection of control penalty.

Table D-1 lists the closed-loop modes that result from the application of the designed GPC PSS. The controller was also found to be stable, having poles at z = 0.9950 and z = 0.5088, thereby meeting the last requirement for any design. The diagram of figure D-2 shows a block diagram of the designed GPC PSS.

Eigenvalues	Damping Ratio (%)	Frequency (Hz)
0.99269 + 0.080586i	4.9961	1.2908
0.99269 - 0.080586i	4.9961	1.2908
0.99463	100	0
0.98458	100	0
0.83459	100	0
0.70903	100	0
0.15516	100	0

Table D-1: Table of closed-loop modes after application of GPC PSS

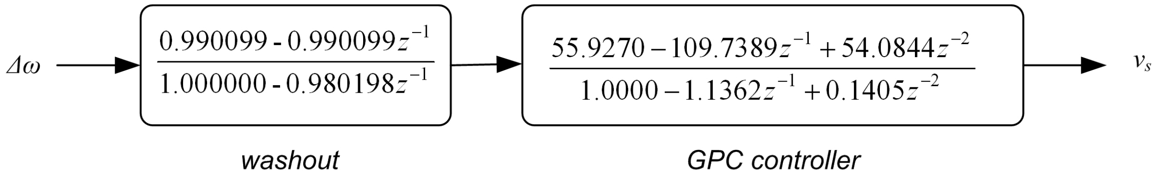
$\Delta\omega$
$\frac{0.990099 - 0.990099z^{-1}}{1.000000 - 0.980198z^{-1}}$
$\frac{55.9270 - 109.7389z^{-1} + 54.0844z^{-2}}{1.0000 - 1.1362z^{-1} + 0.1405z^{-2}}$
v_s
washout
GPC controller

Figure D-2: Block diagram of the designed GPC PSS.

APPENDIX E

Parameter Nudging

The cost function optimized by the GPC algorithm makes no explicit provision for stability. The optimization is only concerned with regulating some process variable to a specified reference value over some short interval of time. It is therefore possible for the GPC algorithm to return a controller that leads to unstable closed-loop process, albeit only marginally so. In the course of trying to resolve the problem of marginal instability a technique labeled *parameter nudging* was formulated. Although the method was eventually abandoned because *marginal instability* could be resolved using simpler techniques, a summary of the parameter nudging technique is provided.

Parameter nudging involves slight modification of the coefficients of the optimally calculated controller in order to make the closed-loop process stable. The modifications were achieved using what can best be described as an *iterative gradient technique* based on *Vieta's Formulas*. [11] A brief summary of Vieta's Formulas is provided followed by a description the parameter nudging method. Finally, details regarding application of the method to resolve the problem of marginally instability are provided.

Vieta's Formulas:

Vieta's formulas [11] define a relationship between the coefficients and the roots of a polynomial of the form given in E-1.

$$a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \cdots + a_n z^{-n} \qquad \text{[E-1]}$$

For the 3rd order polynomial three different equations can be written relating the polynomial coefficients to the poles. They are given in equations E-2 through E-4.

$$s_1 = \sum_{i=1}^{N} r_i = -\frac{a_1}{a_0} \qquad \text{[E-2]}$$

$$s_2 = \sum_{\substack{i,j \\ i<j}}^{N} r_i r_j = \frac{a_2}{a_0} \qquad \text{[E-3]}$$

$$s_3 = \sum_{\substack{i,j,k \\ i<j<k}}^{N} r_i r_j r_k = -\frac{a_3}{a_0} \qquad \text{[E-4]}$$

A 4th order polynomial would add the equation given in E-5. Additional equations of the same pattern would be added for higher order polynomials.

$$s_4 = \sum_{\substack{i,j,k,l \\ i<j<k<l}}^{N} r_i r_j r_k r_l = \frac{a_4}{a_0} \qquad \text{[E-5]}$$

Parameter Nudging Methodology:

The best means of illustrating the developed nudging technique is through a simple example. Take the simple 3rd order polynomial given in equation E-6 wher a_0 is assumed to be unity.

$$1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} \qquad \text{[E-6]}$$

Application of *Vieta's formulas* enable writing of equations E-7 through E-9.

$$r_1 + r_2 + r_3 = -a_1 \qquad \text{[E-7]}$$

$$r_1 r_2 + r_1 r_3 + r_2 r_3 = a_2 \qquad \text{[E-8]}$$

$$r_1 r_2 r_3 = -a_3 \qquad \text{[E-9]}$$

Given these relations the aim is to find the rate of change of each of the roots r_1, r_2 and r_3 as function of the polynomial coefficients. Finding such realtions explicitly is not a trivial task and the computational complexity increases as the degree of the polynomial grows. Linearizing techniques are used to overcome this challenge. Using only the first partial derivatives of the multivariable Taylor series expansion given in E-10 the LHSs of the equations above can approximated.

$$f(x,y,z) \approx f(x_0,y_0,z_0) + \Delta x \frac{\partial f(x,y,z)}{\partial x} + \Delta y \frac{\partial f(x,y,z)}{\partial y} + \Delta z \frac{\partial f(x,y,z)}{\partial z} \qquad \text{[E-10]}$$

Linearizing *Vieta's Equations* gives

$$a_1 = a_1^0 - \Delta r_1 - \Delta r_2 - \Delta r_3 \qquad \text{[E-11]}$$

$$a_2 = a_2^0 + \Delta r_1\left(r_2^0 + r_3^0\right) + \Delta r_2\left(r_1^0 + r_3^0\right) + \Delta r_3\left(r_1^0 + r_2^0\right)\ a_1 = a_1^0 - \Delta r_1 - \Delta r_2 - \Delta r_3 \qquad \text{[E-12]}$$

$$a_3 = a_3^0 - \Delta r_1\left(r_2^0 r_3^0\right) - \Delta r_2\left(r_1^0 r_3^0\right) - \Delta r_3\left(r_1^0 r_2^0\right)\ a_1 = a_1^0 - \Delta r_1 - \Delta r_2 - \Delta r_3 \qquad \text{[E-13]}$$

These equations can then be rewritten as a matrix equation

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \end{bmatrix} + \begin{bmatrix} a_1^0 \\ a_2^0 \\ a_3^0 \end{bmatrix} \qquad \text{[E-14]}$$

$$\begin{bmatrix} \Delta a_1 \\ \Delta a_2 \\ \Delta a_3 \end{bmatrix} = \begin{bmatrix} & & \\ & \mathbf{C} & \\ & & \end{bmatrix} \begin{bmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \end{bmatrix} \qquad \text{[E-15]}$$

where

$$c_{11} = c_{12} = c_{13} = -1$$

$$c_{21} = r_2^o + r_3^o \qquad c_{22} = r_1^o + r_3^o \qquad c_{23} = r_1^o + r_2^o$$

$$c_{31} = -r_2^o \cdot r_3^o \qquad c_{32} = -r_1^o \cdot r_3^o \qquad c_{33} = -r_1^o \cdot r_2^o$$

Equations E-14 and E-15 give the linearized relation defining how changes in a polynomial's coefficients affect its roots. Unfortunately, they are approximate relations that are valid only in the vicinity of the original polynomial and its roots. Still, using the iterative steps listed below the poles can be adjusted as desired.

1. Calculate the linearized relation equation E-14 that relates a polynomial's coefficients to its roots.
2. Specify the desired change to the intended root(s) and calculate the change in the polynomial coefficients needed to achieve that/those changes. The change must be small. Use a faction of the total desired change if necessary.
3. Modify the polynomial coefficients according to the results of step 2 and recalculate equation E-14 using the modified polynomial.
4. Return to step 2 so long as the polynomial's pole requires further adjustment.

Application:

Having described the nudging technique, the next step is to detail its application in solving the problem of marginal instability. Figure E-1 shows how the transfer function of a process and its controller can be collapsed into a single transfer function.

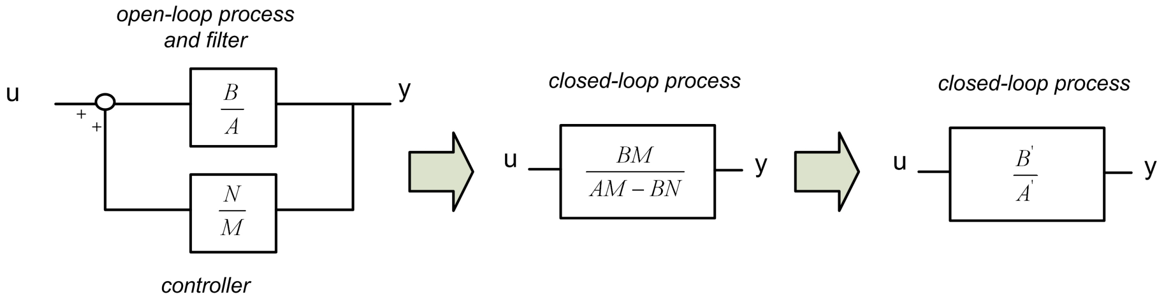

Figure E-1: Calculation of closed-loop transfer function from transfer functions of the controller and the open loop process.

The coefficients of A' cannot be nudged directly since this implies modification of the process. Only the controller parameters can be modified. The first step to getting around this problem is to write an equation for $AM - BN$.

Given, for example, that the polynomials are defined as given in equation set E-16, the coefficients of B' for the indicated powers the shift variable are given in equation set E-17.

$$\begin{aligned} A &= a_0 + a_1 z^{-1} + a_2 z^{-2} \\ B &= b_0 + b_1 z^{-1} + b_2 z^{-2} \\ M &= m_0 + m_1 z^{-1} + m_2 z^{-2} \\ N &= n_0 + n_1 z^{-1} + n_2 z^{-2} \end{aligned} \qquad \text{[E-16]}$$

$$\begin{aligned} z^0 &\Rightarrow \quad a_0 m_0 - \left(b_0 n_0\right) \\ z^{-1} &\Rightarrow \quad a_0 m_1 + a_1 m_0 - \left(b_0 n_1 + b_1 n_0\right) \\ z^{-2} &\Rightarrow \quad a_0 m_2 + a_1 m_1 + a_2 m_0 - \left(b_0 n_2 + b_1 n_1 + b_2 n_0\right) \\ z^{-3} &\Rightarrow \quad a_1 m_2 + a_2 m_1 - \left(b_1 n_2 + b_2 n_1\right) \\ z^{-4} &\Rightarrow \quad a_2 m_2 - \left(b_2 n_2\right) \end{aligned} \qquad \text{[E-17]}$$

Equations set E-17 can be expresses as the matrix equation given in E-18

$$\begin{bmatrix} a_0' \\ a_1' \\ a_2' \\ a_3' \\ a_4' \end{bmatrix} = \begin{bmatrix} m_0 & m_1 & m_2 & -n_0 & -n_1 & -n_2 \end{bmatrix} \begin{array}{c} \begin{array}{ccccc} z^0 & z^{-1} & z^{-2} & z^{-3} & z^{-5} \end{array} \\ \begin{bmatrix} a_0 & a_1 & a_2 & 0 & 0 \\ 0 & a_0 & a_1 & a_2 & 0 \\ 0 & 0 & a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 \end{bmatrix} \end{array} \qquad \text{[E-18]}$$

Here, a$_0$', a$_1$', a$_2$', a$_3$' and a$_4$' are coefficients of the closed-loop transfer function's denominator, A'. Construction of this matrix equation clearly separates those parameters contributing to the closed-loop poles that can be modified from those that cannot. Using the same procedure employed to obtain equation E-14, an equation relating the coefficients of A' to its poles can be written s in equation E-19.

$$\begin{bmatrix} a_0' \\ a_1' \\ a_2' \\ a_3' \\ a_4' \end{bmatrix}^T = \left[\mathbf{C}\right] \begin{bmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \\ \Delta r_4 \\ \Delta r_5 \end{bmatrix} + \begin{bmatrix} a_{00}' \\ a_{10}' \\ a_{20}' \\ a_{30}' \\ a_{40}' \end{bmatrix} \qquad \text{[E-19]}$$

Combining equations E-18 and E-19 results in equation E-20.

$$\left(\begin{bmatrix} m_0 & m_1 & m_2 & -n_0 & -n_1 & -n_2 \end{bmatrix} \begin{array}{c} \begin{array}{ccccc} z^0 & z^{-1} & z^{-2} & z^{-3} & z^{-5} \end{array} \\ \begin{bmatrix} a_0 & a_1 & a_2 & 0 & 0 \\ 0 & a_0 & a_1 & a_2 & 0 \\ 0 & 0 & a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 \end{bmatrix} \end{array} \right)^T = [\mathbf{C}] \begin{bmatrix} \Delta r_1 \\ \Delta r_2 \\ \Delta r_3 \\ \Delta r_4 \\ \Delta r_5 \end{bmatrix} + \begin{bmatrix} a'_{00} \\ a'_{10} \\ a'_{20} \\ a'_{30} \\ a'_{40} \end{bmatrix} \qquad \text{[E-20]}$$

Ideally, the desired change in the problem pole would be specified and then the required change to the controller's parameters would be solved for. This, however, isn't possible because it would require the inversion of the non-square matrix. Instead a trial and error approach is used. The coefficients of the GPC controller are modified one at a time by some fixed percentage of their current value. A series of 'Test vectors' that take the form given in E-21 are generated. Here Δ indicates a small incremental change in the parameter.

$$\left.\begin{array}{l} \begin{bmatrix} m_0 & m_1 \pm \Delta & m_2 & -n_0 & -n_1 & -n_2 \end{bmatrix} \\ \begin{bmatrix} m_0 & m_1 & m_2 \pm \Delta & -n_0 & -n_1 & -n_2 \end{bmatrix} \\ \begin{bmatrix} m_0 & m_1 & m_2 & -n_0 \pm \Delta & -n_1 & -n_2 \end{bmatrix} \\ \begin{bmatrix} m_0 & m_1 & m_2 & -n_0 & -n_1 \pm \Delta & -n_2 \end{bmatrix} \\ \begin{bmatrix} m_0 & m_1 & m_2 & -n_0 & -n_1 & -n_2 \pm \Delta \end{bmatrix} \end{array} \right\} \textit{Test Vectors} \qquad \text{[E-21]}$$

Applying each of these test vectors individually, the LHS of E-20 can be calculated. Solving for the resulting vector of changes to the closed loop poles then becomes a trivial matter. The control parameter change that triggers the most desirable change in the closed loop poles should be implemented. The closed-loop transfer function and subsequently equation E-20 can then be recalculated and the process can be restarted using new 'Test Vectors'. These steps can be repeated until the problematic closed loop pole has be satisfactorily relocated. In this fashion an iterative method suitable for resolving the problem of marginal instability is established.

References

[1] Kundur, Prabha. 1994. *Power System Stability and Control*. New York: McGraw-Hill

[2] Wellstead, P.E., and M.B. Zarrop. 1991. *Self-Tuning Systems: Control and Signal Processing*. London: J. Wiley

[3] Clarke, D.W., C. Mohtadi, and P.S. Tuff. 1987. Generalized Predictive Control – Part I. The Basic Algorithm. *Automatica* 23 (2): 137-148

[4] Clarke, D.W., C. Mohtadi, and P.S. Tuff. 1987. Generalized Predictive Control – Part II. Extensions and Interpretations. *Automatica* 23 (2): 149-160

[5] Proakis, J.G. and D.G. Manolakis. 1996. *Digital Signal Processing: Principles, Algorithms, and Applications*. New Jersey: Prentice-Hall

[6] Barreiros, J A L, A S e Silva, and J A Simões Costa. 1998. A self-tuning generalised predictive power system stabilizer. *Electrical Power & Energy Systems* 20 (3) 213-219

[7] Sung, Su Whan, and J H Lee. 2005. Pseudo-random binary sequence design for finite impulse response identification. *Control Engineering Practice* 11 (8): 935-945

[8] Phillips, Charles L., and H. Troy Nagle, Jr. 1994. *Digital Control System Analysis and Design*. New Jersey: Prentice-Hall

[9] Hansanović, Amer et al. 2004. Practical Robust PSS Design Through Identification of Low-Order Transfer Functions. *IEEE Transactions on Power Systems* 19 (3): 1492-1500

[10] Yadaiah, N., A. Ganga Dinesh Kumar, and J.L. Bhattacharya. 2004. Fuzzy based coordinated controller for power system stability and voltage regulation. *Electric Power Systems Research* 69 (2-3): 169-177

[11] Eric W. Weisstein. Vieta's Formulas. From Mathworld—A Wolfram Web Resource. Retrieved January 3rd, 2007 from the World Wide Web: http://mathworld.wolfram.com/VietasFormulas.html

[12] Signal-to-noise ratio. Retrieved January 3rd, 2007 from the World Wide Web: http://en.wikipedia.org/wiki/Signal-to-noise_ratio

[13] The Mathworks. 2004. Matlab & Simulink: Student Version [Ver. 7.0.1.15 (R14)]

[14] Jackson, G., and U D Annakkage. 2006. Real-Time System Identification for Controller Design. *In Proceedings of the 10th World Multi-Conference of Systemics, Cybernetics and Informatics*. Orlando, USA. July 16-19, 2006

[15] Ling, K.V., and K.W. Lim. 1996. A State Space GPC with Extensions to Multirate Control. *Automatica* 12 (7): 1067-1071

[16] Amano, M, M. Watanabe, and M. Banjo. Self-testing and self-tuning of power system stabilizers using Prony analysis. *Power Engineering Society 1999 Winter Meeting, IEEE*. New York, USA. January 31st, 1999 - February 4th, 1999

[17] Sheng, Jie, Tongwn Chen, and S.L. Shah. 2001. On stability robustness of dual-rate generalized predictive control systems. *Proceedings of 2001 American Control Conference*. Arlington, VA, USA. June 25th-27th, 2001.

[18] Neuman, C.P. 1993. Properties of the delta operator model of dynamic physical systems. *IEEE Transactions on Systems, Man and Cybernetics* 23 (1): 296-301.