UNIVERSITY OF CALGARY

H_∞ Model Predictive Control - Theory and Application

by

Antal Soós

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF ELECTRICAL AND COMPUTER

ENGINEERING

CALGARY, ALBERTA

April, 2005

UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "H_∞ Model Predictive Control - Theory and Application" submitted by Antal Soos in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

_____

Supervisor, Dr. Om P. Malik
Department of Electrical and Computer Engineering

_____

Dr. Ed Nowicki
Department of Electrical and Computer Engineering

_____

Dr. David Westwick
Department of Electrical and Computer Engineering

_____

Dr. Qiao Sun
Department of Mechanical and Manufacturing Engineering

_____

External Reader, Dr. Tongwen Chen
Department of Electrical and Computer Engineering
University of Alberta

Date: 2005 04 15 Calgary,

Abstract

Future industrial systems will require control systems to be more reliable, autonomous, robust, and yet efficient. The emphasis of this thesis is to introduce a robust control algorithm definition that addresses the needs of future industrial environments. The proposed controller is based on an adaptive concept with a two-step approach. In step one, the system model is identified in a closed-loop by a robust technique. In step two, the obtained system model from step one is used to formulate a robust controller.

The system identification is the central part of the controller design since the controller can only be as good as the model that is used to design it. In order to improve the performance and robustness of the system identification, this thesis proposes expert system supervised multiple system identifications. The role of the expert system is to periodically evaluate the estimated models and to propose one for the controller design.

The robust controller is formulated by the H_∞ (sub)optimal design procedure using the proposed system model. The idea behind this controller design technique is to combine an on-line identification algorithm with a control design method that yields a time-varying controller which follows the changing plant.

The effectiveness of the proposed robust controller in an industrial environment is demonstrated by simulation and experimental tests. The proposed robust controller as a power system stabilizer has been tested by simulations on a power system model and in the experimental environment using the micro-synchronous generator at the University of Calgary.

Acknowledgements

I would like to take this opportunity to thank all those people who have helped me along the path to the completion of this research thesis.

I am deeply grateful to my supervisor, Dr. O. P. Malik, for his guidance and encouragement. His enthusiasm and insightful criticism has been inspiring and motivating throughout this study. Dr. Malik has been a wonderful mentor and he was the one who directed my attention to the concept of a multiple model system identification. This idea has determined the direction of my research and added another dimension to the final product.

Some colleagues and friends have been particularly kind in their willingness to impart ideas and information relevant to my research. I would like to acknowledge Dr. Ramakrishna Gokaraju for his help in decoding the cryptic language of power engineers. Mr. Zsigmond Poda supported me with his ideas stemming from his in depth knowledge and expertise in the electrical design of embedded controllers. I am also indebted to Mr. Howard R. Patton, a family friend, who has devoted hours of meticulous reading to the editing of my drafts.

Research of this kind could not have been completed without the assistance provided by the technical support team at Texas Instruments Inc. Their prompt replies to my inquiries and professionalism helped me through the process of transferring developed software to the experimental environment.

Finally, my deepest thanks goes to my family for their understanding and patience. They have tolerated many late nights and lost weekends. I would not have completed this thesis without them.

Table of Contents

List of Tables

List of Figures

Operators and Notational Conventions

General notation practices.

$\square_i$	in general, subscript i refers to the time $t = iT$
$\square_n$	subscript n refers to the present time $t = nT$
$\square^{\mathrm{k}}$	superscript k refers to the element
$\{\circ\}$	sequence with T_s sampling $n = 0, 1, \ldots N$
$(-1, j0)$	complex number, $j = \sqrt{-1}$
$f(x)$	function of argument x
$\|\circ\|_p$	p-norm, p = 1, 2 and ∞
rank(A)	rank of matrix A
Ric( )	Riccati recursion
mse()	mean square error
inf	infimum operator is the greatest lower bound. It is lower than the value under consideration, example: $inf\{1, 1/2, 1/3, \ldots\} = 0$; $inf\{0, 1, 1/2, 1/3, \ldots\} = 0$

sup	supremum operator is the least upper bound
min	minimum operator, example: $min\{1, 1/2, 1/3, \ldots\} = \varnothing$; $min\{0, 1, 1/2, 1/3, \ldots\} = 0$
max	maximum operator,
$\delta_{n,j}$	Kronecker's delta operator ($\delta_{n,j} = 0$ if $j \neq n$)
A^T	transpose of matrix A
A^{-1}	inverse of matrix A
$\aleph$	integer numbers
H_2	optimizes the least squares of the error
H_∞	H_infinity norm, optimizes the maximum energy gain from the error
l_p	p^{th} norm in linear space
l_2	space of square sumable vector sequence over the nonnegative integers
$O(d^2)$	computation complexity of an algorithm
$p(\lambda)$	eigenvalues of A, the m roots of the characteristic polynomial, $p(\lambda) = det(\lambda I - A)$
$\rho(A)$	spectral radius, the maximal modulus of the eigenvalues, $\rho(A) = \max_{1 \leq i \leq m} \lvert\lambda_i\rvert$

$P > 0$	for $P \in \Re^{p \times p}$, means that P is a symmetric and positive definite, $p(\lambda) > 0$
q^{-1}	backward shift operator
$\Re^{p \times q}$	real numbers, real matrix dimension pxq (p rows, q columns)
$\Im$	complex numbers ($\Re$, $i\Re$)
$\forall$	for all

Abbreviations and acronyms.

A/D	Analog to Digital converter
A2D2A	analog interface card: A/D and D/A converter
AC	Alternating Current
ACC	American Control Conference
APSS	Adaptive PSS
AR	Auto Regressive model
ARMA	Auto Regressive Moving Average model
ARMAX	Auto Regressive Moving Average model with an eXogenous signal
AVR	Automatic Voltage Regulator
CPSS	Conventional Power System Stabilizer
D/A	Digital to Analog converter
DARE	Discrete-time Algebraic Riccati Equation
DC	Direct Current, constant part of a signal

DSK	DSP Starter Kits
DSP	Digital Signal Processor
EWMA	Exponentially Weighted Moving Average
FIR	Finite Impulse Response
FQR	Recursive Fast QR RLS algorithm
Gi	Generator number i
GPC	Generalized Predictive Control
KF	Kalman Filter
KFA	modified version of the Kalman Filter
IEEE	Institute of Electrical and Electronics Engineers
IQR-RLS	Inverse QR estimation algorithm
IQR	Inverse QR RLS algorithm
Li	electrical Load number i
LED	Light Emitting Diode
LFT	Fractional Transformation
LMS	Least Mean Squares algorithm
LQ	Linear Quadratic controller
LQG	Linear Quadratic Gaussian controller
LQGR	Linear Quadratic Gaussian Regulator
LQR	Linear Quadratic Regulator
LS	Least Squares

LTI	Linear Time Invariant process
MA	Moving Average mode
MIMO	Multiple-Input/Multiple-Output
MPC	Model Predictive Controller
NoC	No Controller $u_n = 0$
PID	Proportional Integral Derivative controller
PLD	Programable Logic Device
PRO	Proposed model by the expert system
PSS	Power System Stabilizer
QL	algorithm produces a lower triangular matrix Q where the Left of the matrix is non zero
QR-LMS	QR based LMS algorithm
QR	algorithm produces an upper triangular matrix Q where the Right of the matrix is non zero
QR-RLS	conventional QR estimation algorithm
RHC	Receding-Horizon Control
RLS	Recursive Least Squares algorithm
SISO	Single-Input/Single-Output
TCR	Time Constant Regulator
USA	United States of America
WCE	Worst-Case Estimation algorithm

ZOH Zero-Order-Hold

Symbols used in text.

This list contains symbols that have some global use. Some of the symbols may have another local meaning.

A_n state transition matrix

$\bar{A}$ intermediate matrix for the H_∞ optimization algorithm

$\tilde{A}$ intermediate matrix for the H_∞ optimization algorithm

B_n input gain matrix

C_n measurement matrix

D weighting matrix in H_∞ controller optimization

D_i damping coefficient (representing the mechanical as well as the electrical damping effect) for the power system

E weighting matrix in H_∞ controller optimization

E_0 infinite-bus voltage

F^u control signal gain matrix of the state estimator for the H_∞ controller

F^x state transition matrix of the state estimator for the H_∞ controller

F^y system output gain matrix of the state estimator for the H_∞ controller

G	weighting matrix in H_∞ controller optimization
$G(t)$	transfer function of the unknown physical system
$G_{T_s}(t)$	continuous time system observed in T_s intervals
$\hat{G}(\theta, T_s)$	discrete transfer function for system dynamics
H	weighting matrix in H_∞ controller optimization
$H(\theta, T_s)$	discrete error model
H_n^k	information space of the player
h_n^k	state-measurement (-observation) equation, $y_n^k = h_n^k(x_n)$
$I_d(T_K, \gamma)$	entropy integral of $T_K^{\sim}(e^{j\omega}) = T_K^*(e^{-j\omega})$
J^i	cost function
$J(u, \omega, x)$	cost function for two-player zero-sum game
$\bar{J}$	upper value of the cost function for two-player zero-sum game
$\underline{J}$	lower value of the cost function for two-player zero-sum game
J^*	optimal solution for the game
$K_{RLS,\, n}$	RLS algorithm gain
K	players' set, $K = \{1, \ldots, k, \ldots, K\}$, used to distinguish between the actors in the game

K^x	controller gain matrix for the H_∞ controller
L	intermediate matrix for the H_∞ optimization algorithm
L^k	cost functional of player
L^2	space in which the signal is square sumable and the sum is finite
l_1	average of the signal's absolute value in linear space
l_∞	maximum value from the linear space
$l(R^N)$	linear space of a signal observed at N data point
$L(\hat{x}_n)$	information state
$L^k(\theta_n^k, \phi_n)$	performance index for the k^{th} estimation algorithm
L_Σ^k	reminiscence, $R(x_n)$, of the k-th adaptive algorithm
m_d	constant (DC) model order: 0 or 1
m_a	autoregressive (AR) model order
m_b	moving average (MA) model order
$M(\theta)$	estimated system model
N	data sequence size
N	stages of the game, $N = \{0, \ldots, n, \ldots, N\}$, where N is the maximum possible number of moves by a player

o	zero vector
O	zero matrix
P^k	player
pu	per unit a normalized value used in power engineering
P^C	solution matrix of the algebraic Riccati equation for the controller
$\overline{P_{n+1}^F}$	intermediate matrix for the H_∞ optimization algorithm
$\overline{P_n^C}$	intermediate matrix for the H_∞ optimization algorithm
P^F	solution matrix of the algebraic Riccati equation for the state estimation
$P_{RLS,\,n}$	RLS estimated error covariance of the system parameter vector
P_e	electrical power
P_{Mi}	turbine mechanical power applied to rotor
Q	intermediate matrix for the H_∞ optimization algorithm
Q_n	process-noise matrix at the KF
Q_∞	free parameters at admissible controllers to form the set of all stabilizing controllers
$\overline{Q}$	intermediate matrix for the H_∞ optimization algorithm
r_n	reference signal for system operation
$R(x_n)$	reminiscence function

$R_{RLS,\,n}$ estimated error variance of the predicted system output

R_n covariance matrices for measurement-noise at the KF

R intermediate matrix for the H_∞ optimization algorithm

S intermediate matrix for the H_∞ optimization algorithm

S true system

$s_n \in R^q$ desired system output at the KF

$T_{\omega i}$ impulse moment of the rotor

T_K plant with the closed-loop control system, transfer function from $\omega_n \to z_n$

T_i strategy for identification

T_s sampling interval

$T(u_{n-1}, y_{n-1}, \hat{x}_n)$ system state estimation function

u_n discrete time control variable

U_n^k control set of player P^k at stage n

(u^*, ω^*) saddle-point solution or noncooperative equilibrium

$V_n(x_n)$ value function of the game

V_t generator's terminal voltage

$V_n(\theta_n, v_n)$ discrete Lyapunov function for model stability analysis

$\{\nu_n\}$	residual sequence at the system identification
v_n^x	additive disturbance acting on the output of the system
V	intermediate matrix for the H_∞ optimization algorithm
W_θ, W_v	weighting factors
W	space where the disturbance trajectory lies
W	intermediate matrix for the H_∞ optimization algorithm
$\tilde{W}$	intermediate matrix for the H_∞ optimization algorithm
X	state set (space) of the game
X_f	terminal constraint set
X^C	initialization matrix of the algebraic Riccati equation for the controller
X^F	initialization matrix of the algebraic Riccati equation for the state estimation
x_n	state variable vector in discrete time
y_n	discrete time measurement variable, system output
Y_n^k	observation set
Y	space where the measurements trajectory lies
y_n^k	measurement, measurement equation, $y_n^k = h_n^k(x_n)$
Z^N	observed discrete data sequence

z_n	discrete time regulated signal
Δ^F	intermediate matrix for the H_∞ optimization algorithm
Δ^C	intermediate matrix for the H_∞ optimization algorithm
Δ^S	intermediate matrix for the H_∞ optimization algorithm
δ_i	phase angle lead (load angle)
δ	small positive constant/magnitude bounds at WCE
γ	closed-loop norm bound of the H_∞ controller
γ^*	infimum, the lower limit on the minimum achievable norm for γ
Γ_n^k	strategy set (space) of the player
Γ_{n+1}^C	intermediate matrix for the H_∞ optimization algorithm
Γ_n^F	inverse of the innovation variance matrix for the H_∞ controller
Π^F	solution of the Lyapunov equation, matrix
Π^C	solution of the Lyapunov equation, matrix
Δ	model uncertainty $\Delta = \hat{G}(\theta, T) - G_T(t)$
η_n^k	information structure of the game.
θ^0	"true system" parameters
θ_n	linear model parameters $\theta_n = [a_1 \ldots a_{m_a}, b_1 \ldots b_{m_b}, c_1 \ldots c_{m_c}]^T$

Θ_R rotation matrix in QR decomposition

Θ model uncertainty set

$\theta_{\nabla MAX}$ bound on the variable θ_{∇}

ϕ_n observation vector for the linear model

$$\phi_n = [-y_{n-1}, \ldots -y_{n-m_a}, u_{n-1}, \ldots u_{n-m_b}, v_{n-1}, \ldots v_{n-m_c}]^T$$

$\kappa_2, \kappa_\infty, \kappa_\sigma, \kappa_\Sigma$ positive weighting coefficients

Λ_{n+1}^C intermediate matrix for the H_∞ optimization algorithm

Λ_n^F intermediate matrix for the H_∞ optimization algorithm

λ forgetting factor in estimation algorithm, $0 \ll \lambda_R \leq 1$

μ_{LMS} learning rate, adaptation constant at the LMS algorithm

$\bar{\mu}_{LMS}$ learning rate, adaptation constant at the LMS algorithm

$\{\rho_n\}$ positive weighting sequence at WCE

$\Sigma_K(T)$ feedback controller $\Sigma_K : u_n = K \cdot y_n$

ϕ_n measurement vector

Φ_n correlation matrix at QR algorithm

$\Delta\omega$ generator's rotational speed deviation

ω_n inputs to the model which interpret the effects of the model uncertainty

1 Introduction

Many industrial systems exhibit unsatisfactory natural behavior. These systems are characterized by inadequate or unstable performance. In order to compensate for the unsatisfactory natural behavior and to provide the desired stability and performance, a controller shall be implemented. It is expected that the controller will enhance the efficiency and operation of the system by improving setpoint tracking and disturbance attenuation. In this chapter a brief outline of the design process for a robust control system is provided.

1.1 Control

In the past, many control tasks have been successfully addressed by the classical control theory, where the controller is implemented using simple analog technology. Characterized by a certain amount of trial-and-error analysis, this approach to control has matched well the technology of that period. Classical design algorithms, such as lead-lag compensation, were not clearly formalized as design problems to be solved, but rather as tools of practice largely dependent on trial and error processes[1]. With advances in technology and industrial systems, the demand for more adequate controllers has increased. The benefits of implementing more advanced control systems in industry are manyfold.

These benefits include improved product quality, reduced energy consumption, minimization of waste material, and increased system stability.

1.1.1 Modern Control

The first success with adaptive control theory, which addressed the above listed expectations, emerged in 1960s, when Kalman[1] published his ground breaking papers [2] and [3] on linear quadratic feedback control (LQR). The novelty in LQR is the introduction of the state-space representation of the plant as a general linear system and the use of the Riccati[2] differential equation as an algorithm for computing the state feedback gain of the optimal controller with a quadratic performance criterion [4-6].

The first practical implementation of LQR was in the guidance and maneuvering of space vehicles. As it turned out, the new optimal control theory was well suited to many of the control problems that arose from the space program [7]. However, the early success of the LQR was unable to translate into a widespread industrial implementation mainly because of the preliminary assumption that the disturbances affecting the plant can be modelled as stationary stochastic signals with known characteristics. The other weakness of the LQR is that it does not contain a mechanism to include plant model uncertainty in the design. These issues are common in industrial environments. Therefore, a need for a control algorithm which would be less sensitive to plant modelling errors and the lack of statistical information on the disturbances became evident.

---

1. Rudolf Emil Kalman, 1930-, Hungarian born mathematical system theorist
2. Jacopo Riccati, 1667-1754, Mathematician in Venetian Republic

A solution to the above problems was proposed in 1981, when Zames[1] published his pioneering paper [8] on H_∞ control, which initiated the development of robust control theory. The fundamental idea of the H_∞ control theory is that the disturbances effecting the plant can be modelled as deterministic square integrable signals, where only an upper bound on the signal power is assumed to be known. The controller is selected in such a way that it minimizes the disturbance energy gain on the plant output. In other words, the H_∞ control theory addresses the issue of worst-case controller design. The designed controller is robust with respect to model uncertainty and deals with unknown disturbances.

Since the publication of [8], H_∞ control theory has been extensively studied and many different techniques have been used to develop the H_∞ control algorithm [9-18]. Among the various time-domain approaches to this class of worst-case design problems, the one that uses the framework of dynamic (differential) game theory seems to be the most natural [19]. In the game theoretical framework, the H_∞ control problem can be regarded as a game where nature (the opponent) has access to the unknown exogenous input and the designer has a choice for the controller. The objective in selecting the controller is to obtain a design that minimizes a given performance index under the worst possible disturbances or parameter variations in the plant model. At the same time, nature (the opponent) has the objective to maximize the same performance index by controlling the exogenous input by selecting the worst possible disturbances. In the literature, this is described as the

1. George Zames, 1934-1997, Polish born control theorist, McGill University

two player zero-sum game between nature - the maximizing player, and controller - the minimizing player.

The resulting H_∞ controller has a state feedback structure where the gains are calculated by two recursive Riccati equations. One, a backward Riccati recursion, is used to calculate the controller gains. The other, a forward Riccati recursion, is used to calculate the gain for the state estimator.

The control design obtained under these guidelines is an H_∞ controller that is less sensitive to plant model parameter variations and more robust to disturbances than in the case of LQR control. Thus the controller regulates the plant against the worst possible disturbances. However, the resulting controller may be overly conservative [20]. Finally, it is worth noting that the H_∞ control includes LQR control as a limiting case. More precisely, as the specified closed-loop norm bound (normally denoted by γ) tends to infinity, the central solution of the H_∞ control problem converges to the corresponding LQR control [1].

1.1.2 System Identification

In order to implement the H_∞ controller, it is required to obtain a linear model of the plant to be controlled. Since the early 1980s, it has been gradually recognized that the real challenge of the control engineering is the modelling of the plant to be controlled [1]. Originally, the plant model is derived by formulating the physical laws that describe the physical process of the plant. In this approach, to successfully obtain the plant model it

requires not only a thorough understanding of the plant's physical process but also the knowledge of it's parameters. Outside of the research laboratories and space exploration, most plants are too complicated for this type of modelling. In addition, formulating the appropriate physical laws can be time consuming, which makes the method economically not feasible.

These constrains are resolved in the *system identification* approach, where the model is constructed by fitting a parameter model to the observed data. The system identification method is characterized by the fact that the resulting model does not have any physical interpretation, and it describes only the relationship between the observed input(s) and output(s) of the plant.

The process of system identification can be also seen as a search for the solution of an overdetermined system of equations. The *overdetermined system* is characterized by a significantly higher number of equations than variables. In system identification the observed input-output data corresponds to the equations, while the model parameters stand for the variables. An exact solution for the overdetermined system does not exist, only an approximate one. The approximate solution is based on a criterion for solution search. In the literature there are numerous criteria to fit the model parameters to the observed data. The selected criterion defines the identification method.

Generally, the criterion for solution search is based on minimizing a particular norm of the error between the output from the estimated model and the observations. There are three dominant norms in the literature. The first norm criterion is when the criterion is selected to minimize the average absolute value of the difference between the model and the observations. The second norm criterion is applied when the parameters for the model

are selected in a way to minimize the average squares of the difference between the model and the observations. The third or infinity norm criterion is implemented when the goal is to minimize the single largest difference between the model and the observations. These three types of identification methods are the basis for the different types of algorithms. In the research literature, a countless number of implementation methods can be found for identification algorithms.

It is impossible to define the best identification method due to the fact that for the different implementations the criterion for the “best” changes with the particular requirements. A system identification that uses many different models simultaneously dismisses the task of evaluating the best identification method for a particular implementation. Instead, an expert system is designed which, in real-time, selects the best identification method from a pool of methods running in parallel. This real-time multi model system identification is possible due to the advances in computation power.

The controller design that is able to improve the performance of a controlled system by means of system identification and robust control, begins by recognizing the link and interaction between modelling and control [21]. The robust control-oriented identification procedure is a closed-loop identification, where the controller provides relevant information on the dynamics of the plant to feedback control. An interactive scheme for modelling and redesigning of the feedback controllers is described in the following.

1.1.3 Model Predictive Control

If the controller is designed using a system model obtained by system identification, it can be described as parameter-adaptive control. In this approach, the system parameters are estimated in real-time and used for the controller on-line parametrization, as shown in Figure 1-1.

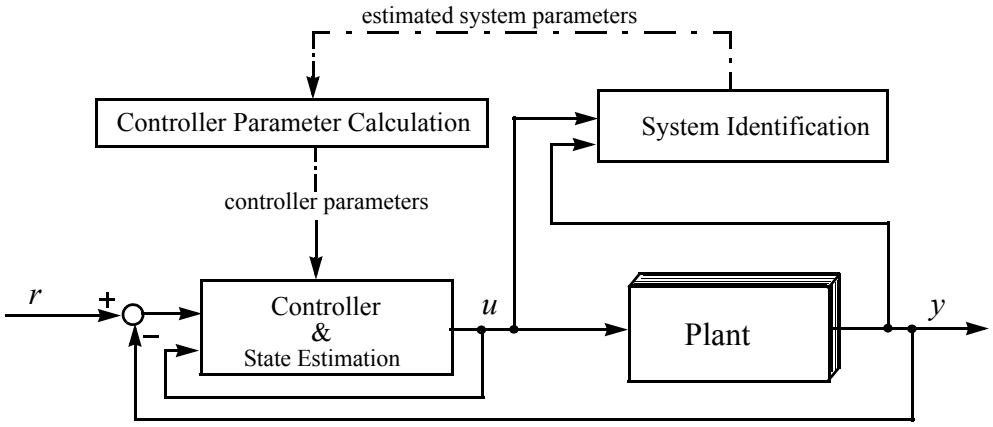


Figure 1-1 Parameter adaptive controller - model predictive controller

Due to the fact that the model with estimated parameters only describes the past observations of the plant's input and output, and it only predicts the future behavior, this type of controller is called *model predictive controller* (MPC). The term MPC does not designate a specific control strategy but rather an ample range of control methods, which make explicit use of a model of the process to obtain the control signal by minimizing the objective function [22]. In other words, the MPC is an optimal control-based strategy that uses a plant model to predict the effects of an input sequence on the evolving state of the plant [23]. In general, the control action of the MPC is calculated by minimizing an objective

function over a given horizon, producing an optimal open-loop control sequence. The first control in that sequence is applied to the plant until another plant measurement becomes available. At the next sampling instant, a new optimal control problem is formulated and solved based on the new measurements [24]. The objective function is defined in terms of both present and predicted system variables and is evaluated using an explicit model to predict future process outputs [25].

Overall, the MPC can be summarized as an explicit use of a model for controller formulation, where the controller is calculated by minimizing an objective function. The resulting controller has a receding strategy characterized by the displacement of the horizon towards the future as new updates from the system become available.

The research in this thesis describes the H_∞ MPC implementation, where the system model is obtained in real-time by system identification processes, and it is used to determine the discrete-time H_∞ (sub)optimal controller, also in real-time. Each time when an updated system model is available, an optimal control problem with an H_∞ control objective is solved. The resulting H_∞ controller is utilized until a new update on the system model becomes available, thereby providing an update to the H_∞ controller.

1.2 Application of the Proposed Approach

A method for designing an H_∞ MPC output feedback controller for complex nonlinear systems is proposed in this thesis. Using this methodology, a power system stabilizer is

implemented. Through the implementation process, the H_∞ MPC is examined to determine how well it handles model uncertainty, disturbances and measurement noise. The results are compared to that of a conventional power system stabilizer (CPSS) on the same benchmark power system model realized in software and on a scaled physical model.

1.2.1 The Power System

Electrical energy has become a major form of energy for end use consumption in today's society. To make electric energy generation and transmission more economic and reliable, the trend in electric power production is towards an interconnected network of transmission lines linking generators and loads into large integrated systems. The power station sites are selected close to a source of energy, such as large coal mines and water power sources. Consequently, power transmission lines and networks have to meet the following requirements [26]:

- connect distant power stations in secure parallel operation
- transmit electrical power to large load centers over large distances
- ensure satisfactory parallel operation with other interconnected power systems.

For proper operation, this large integrated system requires a stable operating condition. Stability in power systems is generally regarded as the ability of the generating units to maintain synchronous operation [26].

1.2.2 Nature of Power System Oscillations

Smaller power systems have hundreds of kilometers of transmission lines; while the largest (the eastern USA/Canadian interconnected system) has thousands of kilometers of transmission lines. Most electric power systems are AC with a frequency that is almost uniform over the whole network. This is achieved by using synchronous AC generators. In small systems, there may be only tens of generators; in large systems there are thousands [27]. System frequency is held within tight limits by governing the speed of the generator prime movers, and system voltages are maintained by generator excitation system control.

Interconnected AC generators produce torques that depend on the relative angular displacement of their rotors. These torques act to keep the generators in synchronism (synchronizing torques). Thus, if the angular difference between generators increases, an electrical torque is produced that tries to reduce the angular displacement. It is as though the generators were connected by torsional springs, and, just as in mass-spring systems, the moment of inertia of the rotors and the synchronizing torques cause the angular displacement of the generators to oscillate following a system disturbance. The angular displacements should settle to values that maintain the required power flows through the transmission network and supply the system load [27].

If the disturbance is large, for example a prolonged three-phase fault on the transmission system, the nonlinear nature of the synchronizing torque may not be able to return the generator angles to a steady state. Some or all generators then lose synchronism and the system exhibits transient instability. On the other hand, if the disturbance is small, the syn-

chronizing torques keep the generators nominally in synchronism, but the generators' relative angles oscillate. In a correctly designed and operated system, these oscillations decay: the system is then called small-signal stable. In an overstressed system, small disturbances may result in oscillations that increase in amplitude exponentially: the system is then said to be small-signal unstable [28].

Unstable power system oscillations have occurred all over the word in the last 30 years. They appear first when a power system is pressed to supply increasing load. As transmission lines are loaded more and more, the generators need to rely more heavily on their excitation systems to maintain synchronism, and at some point, without supplementary control, the synchronizing oscillations become unstable. Also, during the last 30 years, many power systems have been interconnected in order to enable the exchange of power. This allows operating cost to be kept at a minimum.

The interconnecting ties between neighboring power systems, although they may not be overloaded, are often relatively weak when compared to the connections within each system. The synchronizing torques are lower across these weak ties; and this, coupled with the high aggregate inertia of each of the systems being interconnected, leads to low frequency inter-area oscillations. Many of the early instances of oscillatory instability occurred at low frequencies when these interconnections were made [27].

1.2.3 Power System Stability Modelling

Stability calculation methods have always lagged behind interconnected power system size, so there has been a continuous striving for suitable simplifications. A kind of

"instinctive simplification" was to divide the theoretically unique problem of stability into two parts: steady-state and transient stability, for which analysis methods have been developed independently over many years. In the American literature, the steady-state stability and dynamic stability concept pair is in use. The analytical approach to a synchronous machine's fundamental behavior is best modeled by Park's[1] reference frame [29], see Appendix A and B. In order to obtain some appreciation of power system oscillations, it is important to include a discussion of stability analysis. However, this is not the main purpose of this study.

1.2.4 Steady-State Stability

Steady-state stability analysis is the study of a power system and its generators in strictly steady state conditions. It is also an attempt to determine the maximum possible generator load that can be transmitted without loss of synchronism of any one generator. The maximum power is called the steady-state stability limit.

For an *n*-machine power system the active power fed in by the i^{th} generator is defined by the equation

$$P_{Ei} = \frac{U_{pi}^{2}}{Z_{ii}} \sin\alpha_{ii} + U_{pi} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{U_{pj}}{Z_{ij}} \sin(\delta_i - \delta_j - \alpha ij) \tag{1.1}$$

1. R.H. Park, Electrical Engineer, General Electric Company

where U_{pi} is the magnitude of the internal voltage (the voltage behind synchronous reactance) of the i^{th} generator (line to line voltage); $Z_{ii}\left(\frac{\pi}{2}-\alpha_{ii}\right)$ is the driving point impedance "seen" from the internal voltage; $Z_{ij}\left(\frac{\pi}{2}-\alpha_{ij}\right)$ is the transfer impedance between machines *i* and *j*; δ_i is the phase angle lead (load angle) of the i^{th} generator with respect to the reference phasor and P_{Ei} is the electrical three phase power of the i^{th} generator [26].

Assuming that the load angles of all other machines are constant, the study-state stability limit can be predicted from (1.1).

A common problem is the insidious nature of oscillatory instability. Power flow over a tie-line may be increased to supply remote load with no noticeable problems until the stability limit is reached. A slight increase in power flow beyond this limit results in oscillations in which amplitude increases quickly with no need for any system fault. At best, the system nonlinearities limit oscillation amplitude. At worst, the oscillation amplitudes reach levels at which protective relays trip lines and generation, and this in turn causes partial or total system collapse [27].

1.2.5 Dynamic Stability

Dynamic stability is a concept used in the study of transient conditions in power systems. Any electrical disturbance in a power system will cause electromechanical transient processes. Besides the electrical transient phenomena produced, the power balance of the

generating units is always disturbed, and thereby mechanical oscillations of machine rotors follow the disturbance.

To describe the transient phenomena, the well-known swing equation of the synchronous generators, derived from the torque equation for synchronous machine, is used

$$T_{\omega, i}\frac{d^2\delta_i}{dt^2} = P_{M, i} - K_{d, i}\frac{d}{dt}\delta_i - P_{E, i} \tag{1.2}$$

where $T_{\omega, i}$ is "the impulse moment" of the rotor of the generating unit, $K_{d, i}$ is the damping coefficient (representing the mechanical as well as the electrical damping effect), δ_i is the phase angle (load angle), $P_{M, i}$ is the turbine power applied to rotor and $P_{E, i}$ is the electrical power output from the stator [26].

1.2.6 Synchronizing Oscillations

Two types of synchronizing oscillations are common in all interconnected AC power systems. The first is associated with a single generator (or a plant of identical generators) acting against the system. The second is more complex and involves generators in one area of the power system oscillating against other generators in other areas of the power system. Local or plant modes of oscillations have natural frequencies of about 1 to 2 Hz. Inter-area modes of oscillation have lower natural frequencies on the order of 0.1 to 0.7 Hz. In small systems, inter-area oscillations generally have higher natural frequencies than those of large systems [27].

The total number of modes of synchronizing oscillations is equal to one less than the number of interconnected generators. In a system having thousands of generators, there are thousands of modes of oscillation. All of these modes of oscillation must decay following a system disturbance. If any one mode increases in amplitude, the system's operators would have to take action to prevent either local or system-wide collapse.

Power systems must be designed to be stable under as wide a range of system loads and operating conditions as possible. Generally, if the operation of the system is constrained, those constraints should be due to the thermal operating limits of the transmission system or loss of synchronism (transient instability) and not due to oscillatory instability.

To determine the nature of system oscillations, analysis of the following system characteristics is required:

- Frequency and damping of the system’s synchronizing oscillation
- Pattern of generators that take part in each mode of oscillations.

Generators that are able to have a controlling effect on the oscillations must be identified, and tools must be provided to allow for efficient and robust design of oscillation damping controls [27].

1.2.7 Damping Controls

Power system stabilizers are the most cost-effective tools for power system oscillation damping controls. Essentially, they use the power amplification capability of the generators to generate a damping torque in phase with the speed change. This is achieved by

injecting a stabilizing signal into the excitation system voltage reference summing junction as presented in Figure A-2 in Appendix A.2. The stabilizing signal is most often the change in generator rotor speed, phase advanced to counteract the phase lag between the exciter voltage reference and generator electrical torque [27].

Historically, in the late sixties, the need for such devices was demonstrated by some cases of sustained power swings in the western part of the USA. Subsequent analysis has shown that these swings were due to poor damping characteristics caused by modern voltage regulators of conventional structure, but with comparatively high gain from the stability point of view. To compensate for the unwanted effect of these voltage regulators, additional signals were introduced in the feedback loop of voltage regulators. The additional signals were mostly speed deviation, AC bus frequency, or accelerating power. The devices set up to provide these signals through properly chosen transfer functions have been called "*power system stabilizers*" [26].

The basic objective of a power system stabilizer is to modulate the generator's excitation in such a way as to provide additional damping to the electromechanical oscillations of generating units; thereby improving the steady-state stability of the whole power system. To do this the power system stabilizer has to produce a component of electrical torque in the synchronous machine that is in phase with the speed variation of rotors.

Designing a power system stabilizer is a challenging task because of the highly nonlinear dynamics and strong performance requirements of the power system under the wide variations in plant parameters. The present industrial control realization is dominated by the conventional PSS (CPSS) type of control, described in Appendix A.3. The dominance of CPSS can be explained by the simpleness of the lead-lag compensator implementation

and maintenance, using trial and error. However, when a CPSS is used to control a highly nonlinear process, the controller must be tuned very conservatively in order to provide stable behavior over the entire range of operating conditions. Conservative controller tuning can result in serious degradation of control system performance.

There have been problems in the past with power system stabilizers causing steam turbine shaft torsional modes to become unstable, but this risk has been eliminated in many modern power system stabilizer designs. However, there are still many problems with installed power system stabilizers that have been introduced by ineffective commissioning and tuning of the devices. Generally, in systems with both local and inter-area modes, power system stabilizer parameters are determined through off-line analysis, and tuned further during commissioning. The validity of the model used in the off-line studies should be checked on commissioning. Setting power system stabilizers to typical values is particularly dangerous for systems in which inter-area modes are of concern. It is very easy for the stabilizer to have a destabilizing effect at low frequencies that cannot be observed during on-line commissioning tests. Further, it is easy for controller settings to be inadvertently changed due to nonlinear changes in generators' and in transmission-lines' operating conditions. The models based on original manufacturers' information may not be accurate in few years [27]. This problem can be resolved by implementing an H_∞ MPC where the nonlinear changes in generators' and in transmission-lines' operating conditions is tracked by system identification and the controller is adjusted accordingly.

1.3 Thesis Objective

In this research thesis a new control algorithm with a systematic design procedure is proposed. This procedure consists of a robust system identification and an H_∞ suboptimal control law. The resulting controller is an H_∞ MPC algorithm. A detailed implementation process is provided for the H_∞ MPC, using the finite horizon H_∞ (sub)optimal control algorithm derived from the game theory. The H_∞ controller is calculated using two recursive Riccati equations, one for the controller and one for the state estimator. The convergence of these Riccati recursions is dependent upon the initial values given for the recursions. Strict specifications for the convergence conditions are provided using Lyapunov[1] equations.

The objective of this study is to present a systematic methodology for building an H_∞ MPC algorithm, which is ready for application. The difficulty with the implementation of H_∞ MPC is that it requires a sophisticated background in mathematics.

The purpose of this research project is to translate a complex mathematical theory of control into a format that is easy to comprehend, and to identify how H_∞ MPC can be successfully applied in industrial designs. In the test environment, the H_∞ MPC is applied as a power system stabilizer. The role of the applied control algorithm is to solve the power system stabilizer design problem in the face of nonlinear plant and exogenous disturbances.

---

1. Alexander Mikhailovich Lyapunov, 1857-1918, Russian mathematician

By employing digital signal processing techniques more complex control algorithms, such as the H_∞ MPC, can be used in a variety of control devices. It is hoped that this research will make a contribution to the development and application of robust control implementation.

1.4 Contribution of the Thesis and Novelty

The motivation for this research arose from recent developments in robust control and control-oriented identification. The main contribution of this thesis is the development of a robust control-oriented modelling procedure for an unknown plant subsequently followed by a robust control design. The model is obtained by a control-oriented system identification procedure. Although, this model will always be a simplified representation of the unknown plant, the subsequent robust control is designed to account for the presence of the modelling error.

The system identification procedure developed in this thesis addresses the problem of robust identification by using multiple system identification techniques simultaneously and by selecting the best model through an expert system. The best model is the one which most likely describes the plant until the expert system's next decision. In this way, a reliable model for a robust controller design is obtained.

In order to achieve the goal of this thesis, the following research areas have been merged:

- Control relevant system identification of models in closed-loop.

- Expert system design for model selection from the identified set of models.
- Robust control design using model based approach.

Within the above mentioned areas existing results on system identification and robust controller design techniques have been employed. However, in order to define a robust control procedure, the following were developed:

- A new expert system, which selects the model that most likely describes the plant in the future.
- An expert system reminiscence, which is modelled using an exponentially weighted moving average (EWMA) function.
- A definition of system identification stability, by extending the Lyapunov stability criterion with the practical stability criterion.
- An expert system decision using the Bellman's[1] Principle of Optimality.
- The limits on the initial values for the backward and forward Riccati recursions, to be used in the H_∞ (sub)optimal controller optimization algorithm.
- A definition of MPC stability by extending the Lyapunov stability criterion with the practical stability criterion.
- A new algorithmic description for the implementation of the H_∞ (sub)optimal controller.

The contribution of this research is that it provides a proof through simulation and experimental studies that an expert system supervising multiple system identification leads to a robust system identification, which can be used for robust control in the form of

1. Richard Bellman, 1910-1984, American applied mathematician

an H_∞ MPC. The new multi-model H_∞ MPC has successfully been implemented as a power system stabilizer. The application of the robust multi-model identification in combination with an H_∞ control design procedure led to a controller with a guaranteed robust performance.

As a consequence of the development of the new multi-model H_∞ MPC, the following by-products emerged:

- A matrix positivity test algorithm;
- An ARMA model stability test algorithm;
- A median based algorithm to estimate the frequency deviation of the power system.

Also, it has been proven that using the present DSP technology, it is possible to simultaneously calculate seven different types of system identification algorithms, namely, the RLS, LMS, Kalman Filter, three QR based algorithms and the worst-case identification algorithm along with the H_∞ (sub)optimal controller, within 33 ms sampling time. The H_∞ optimization algorithm for the H_∞ (sub)optimal controller weight's calculation and the model selection expert system, however, run once every second.

In order to achieve a real time implementation of the proposed algorithm, an embedded system has been developed. This embedded system was built from two DSP boards and from an analog card and a digital interface card.

The thesis is written with the intention to allow an easy usage and implementation of a H_∞ MPC algorithm following the steps defined in this research.

1.5 Outline of the Contents

This thesis is divided into four parts and 12 chapters. The first two parts contain the theoretical background, while the last two parts present the results from the simulations and experiments. For ease of reference, a brief summary for each of the chapters is provided below.

- **Part I - "System Identification"** This part is divided into Chapters 2 and 3. Chapter 2 describes the underlying theoretical framework used for system identification. In Chapter 3, the idea of a multi-model system identification is introduced in order to implement a robust system identification. A variety of system identification techniques are reviewed, which are used as building blocks for a multi model system identification. In multi model system identification various system identification techniques are implemented in parallel, from which the best is selected periodically in real-time. The selection process for the best system parameter is investigated and an expert system for this parameter selection is described.
- **Part II - "Robust Control"** - This part consists of Chapters 4 - 7. Chapter 4 provides a brief overview of the robust control theory. Chapter 5 gives a general description of two player game theory. The game theory is used as the basic theoretical framework for the H_∞ control algorithm derivation. In Chapter 6, the algorithmic definition for the discrete time H_∞ control algorithm is presented. And finally, in Chapter 7, theory for the H_∞ MPC is reviewed, where the system identification and the H_∞ controller calculation are integrated within the MPC theory.

- **Part III - "Simulation Studies"** - In this part, the simulation studies are described, where the H_∞ MPC is used as a power system stabilizer. The simulation was conducted on a UNIX system based power system model, which was developed at the University of Calgary. Part III is divided into two Chapters: first, the "Single-Machine Infinite-Bus System" simulations are presented followed by the "Multi-Machine Power System" simulations, in Chapters 8 and 9, respectively.
- **Part IV - "Experimental Tests"** - In addition to the theoretical investigations and simulation studies, the behavior of the H_∞ MPC power system stabilizer is tested in real-time, on a physical model of a power system. Chapter 10 describes the test environment in a format of a power system model located in the Power Research Laboratory at the University of Calgary. This power system model, also called a micro machine power system, consists of a scaled synchronous generator and transmission line model. Chapter 11 presents the results of the experiments collected from the micro-machine power system.

Part - I System Identification

The focus of this thesis is directed to the derivation of dynamic models (system identification) to be used for model based (robust) control designs. The aim is to extend previous research in closed-loop identification in robust control. This research is an experimentation with the supervisory (model switching) control, in an effort to ensure good adaptation transients in the presence of large parameter variations.

A robust multi model based system identification is described. The discrete time varying model of the system is constructed by using the observed input/output data. The proposed technique applies an expert system, which periodically selects a set of system parameters from those estimated by different adaptive algorithms. Selection is based on the Principle of Optimality in decision making by analyzing the models' stability. Adaptive algorithms with the H_2 *and* H_∞ *norms of index performance and with a "Worst-Case Estimation" strategy calculate concurrently the discrete models of the system. To improve computational efficiency and numerical stability of the previously mentioned algorithms, the QR decomposition approach is employed. The obtained model is a fundamental and an essential element in the construction of an optimal controller, which has superior adaptation to the system parameter changes.*

2 Introduction to System Identification

In order to characterize the relevant systematic dynamic relations of the system to be controlled, most modern control synthesis methods require some kind of mathematical model. In general, there are two approaches to the system model construction: *white-box* and *black-box* models [30].

The construction of the white-box or physical modelling of the system requires detailed knowledge of the physical laws governing the behavior of the system. In this model, all parameters and variables can be interpreted in terms of physical entities.

In the black-box or experimental approach, a model is constructed from the measured input and output data using no physical insight whatsoever, and the model parameters are simply knobs that can be adjusted to optimize the model fit. This modelling approach, which characterizes any systematic dynamic relations that are present in the system, is based on experimental data observation and is also called *system identification*. Despite the simplistic nature of many black-box models, they are very efficient for modelling dynamic systems and require less engineering time to construct than white-box models [31].

The generic problem formulation for system identification using the black-box model can be described as an observation of the unknown system, $G(t)$, based on the collected N discrete data points, $Z^N = [\{u_n\}, \{y_n\}]$, that consist of the applied input signal, $\{u_n\}$, and the observation of the (possibly) disturbed output signal, $\{y_n\}$, as presented in Figure

2-1 [21]. The *additive disturbance*, $\{v_n^x\}$, acting on the output of the system is used to model the effects on the observation of $\{y_n\}$ that cannot be described by the input $\{u_n\}$, applied to the system [32].

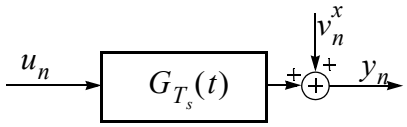


Figure 2-1 Observation of an unknown system

The black-box model approach provides satisfactory results in system identification through its ability to construct models which describe the map from the input to the output of the system, and to use this map in the design of a feedback controller.

2.1 Control Oriented System Identification

Many systems have to operate under feedback control for safety or productivity reasons [33]. Under feedback control, the input and output of the system $Z^N = [\{u_n\}, \{y_n\}]$ can only be observed in the so called closed-loop condition. Further, in order to track the system parameter changes during the time of regular system operation, the system identification should be performed continuously. This practice of persistent system identification under feedback control must overcome the correlation between the system input and output. As presented in Figure 2-2, the system output is correlated to the system input due to

the system's dynamic behavior. On the other hand, in a closed-loop condition the system input is correlated to its output due to the action of the controller

$$y_n = f(G(\theta^0, T_s), u_n) \qquad \text{and} \qquad u_n = f(\Sigma_K(T_s), y_n). \tag{2.1}$$

In order to resolve this problem of correlation affecting the system identification, an external perturbation signal is introduced to the feedback loop. The external perturbation, in Figure 2-2, which may be a zero-mean white-noise signal generated by the controller, shall be un-correlated with the elements of the observed input-output data sequence $Z^N = [\{u_n\}, \{y_n\}]$ [43]. While this external perturbation signal has to sufficiently excite the system for identification, the introduced disturbance should not compromise the plant's required working conditions.

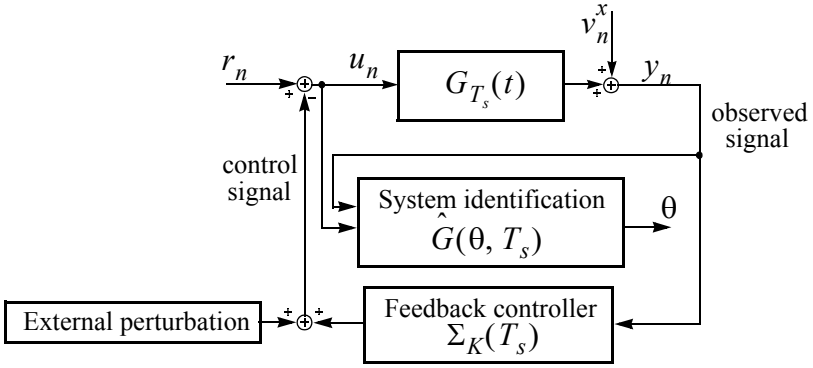


Figure 2-2 Closed-loop system identification

Identification in closed-loop [34-39], as presented in Figure 2-2, leads to a *control oriented identification*, which is a better predictor for the closed-loop dynamics, resulting in a more accurate estimation of the system model, $M(\theta)$, at the control relevant frequencies.

The control relevant frequency range typically corresponds to the frequencies in the neighborhood of the cross-over frequency, on the Bode[1] plot, of the controlled system [40]. In other words, the objective of the system identification in the closed-loop is to achieve better prediction for the closed-loop dynamics via estimation of the system model on the critical frequencies in the region where the Nyquist[2] plot is closest to the critical point $(-1, j0)$ [41], shown in Figure 2-3.

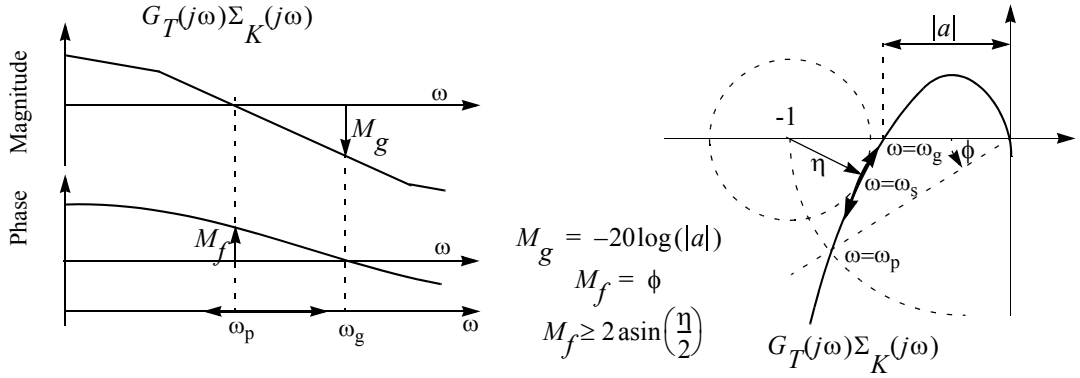


Figure 2-3 Control relevant frequency range

It was found [42] that the models identified in closed-loop are particularly suitable for model-based robust control design because the controller can have a linearizing effect on a nonlinear plant's dynamic behavior at a relevant working point, enabling accurate linear modelling.

1. Hendrik Wade Bode, 1905-1982, Madison, Wisconsin, Electrical Engineer at Bell lab.
2. Harry Nyquist, 1889-1976, Nilsby, Sweden, Electrical Engineer at Bell lab.

2.2 System Identification by Multiple Models

In this thesis, the black-box model based system identification is enhanced by the ideas described in [41] and [44], which suggest a multi model based system identification. According to the literature [41-45], the multiple model based system identification has an advantage when the system's working point changes abruptly due to system error or external disturbances.

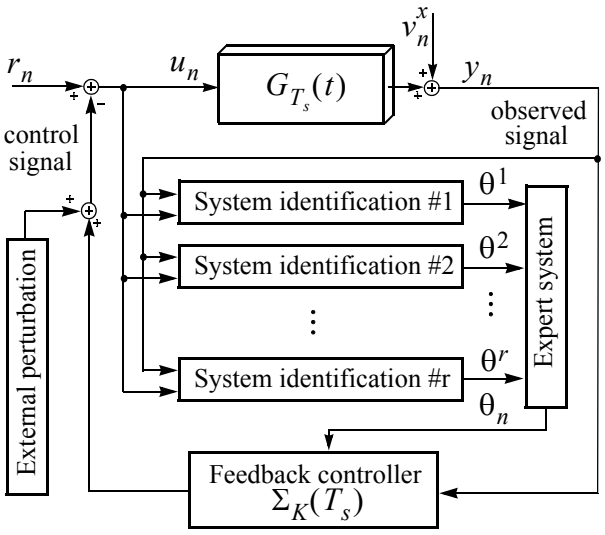


Figure 2-4 Multiple model system identification

The approach proposed here implies that instead of using one adaptive algorithm and several fixed models suggested in earlier studies [41-45], the model of the plant should be estimated with different system identification techniques. The use of diversified parameter estimation techniques will guarantee better estimated parameter accuracy and stability.

This diversification of parameter estimation is made possible by techniques such as H_2, H_∞, Kalman Filter, Worst-Case Estimation, etc. These techniques are concurrently performing the system identification. The selection process evaluates the resulting models and the one, which may show the best performance in the future, is selected as a representation of the process. An elegant solution to perform this selection process is to develop an expert system.

Implementation of the approach suggested in this research requires large processing power for calculations in real time. Recent developments in Digital Signal Processing (DSP) devices, by various manufacturers, provide the necessary processing power for calculating algorithms in industrial environments.

The rest of this chapter is dedicated to establishing a theoretical background for an expert system, which selects the best system identification technique. Special attention is given to model stability which can be obtained using multi model system identification. In the following sections, system identification is defined on an abstract level.

2.3 Model Set Definition

In this research, system identification is based on the assumption that the observed input-output data sequence, $Z^N = [\{u_n\}, \{y_n\}]$ of length N, is generated by the "*true system*" S [46]

$$S : y_n = G(\theta^0, T_s)u_n + H(\theta^0, T_s)\nu_n \tag{2.2}$$

where T_s is the sampling time and $G(\theta^0, T_s)$ is the transfer function, which represents the dynamics of the system. The second part of the equation is the *additive noise model,* sometimes called *residual*, which accounts for both external perturbing forces and unmodelled system dynamics. According to the *Wold decomposition theorem* [47], the additive noise model can be interpreted as a component that is constituted from the deterministic part, $H(\theta^0, T_s)$, and the white-noise sequence, $\{v_n\}$. The deterministic part, $H(\theta^0, T_s)$, models the systematic (predictable) information which is observed on the system output sequence $\{y_n\}$ and cannot be correlated to the system input sequence, $\{u_n\}$.

The so called *true system parameters,* θ^0, where $\theta^0 \in R^d$, are unknown but can be determined through estimation. The estimation procedures are developed by the strive to minimize the norm of the error between the output of the model and the measurements. Different norms render different optimum solutions, with different deficiencies in the estimation process. In order to address the presence of deficiency in the modelling process, the system is not modelled by a single model. Instead, the system is assumed to lie in a *set of models* that is built up from a nominal model along with an allowable model perturbation [21]. These estimated coefficients are denoted by θ^k, where $k = 1, 2\ldots, r$ and it differentiates between the various estimated parameter vectors.

The *estimation* can be described as a procedure where the *model structure*, M, defined with the assumed system S in (2.2), is parametrized by the coefficient vector, θ^k, forming a *set of model*s, M^*, and it is for the estimation procedure to select that member in the set that appears to be the most suitable for the purpose in question [30]. In this way, the esti-

mated parameter vector, θ^k, selects a particular model, $M(\theta^k)$, from the model set M^*, presented in Figure 2-5.

Determining the model structure, M, involves the following steps: first, the type of the model has to be selected (e.g., nonlinear, linear, black-box, white-box, etc.); second, the size of the model needs to be determined (e.g., state-space model order, degrees of the polynomials in the model, etc.); and third, the model parametrization has to be completed in order to find a suitable model structure. There is a freedom in choosing a nominal model, M, upon the assumption that the system S $(M \ni S)$ will supposedly be described through the input/output observations with the given bound on the modelling error. However, both the nominal model, M, and the bound on the modelling error will determine the structure of the model uncertainty set, Θ, and its suitability for the design of a robust controller [30].

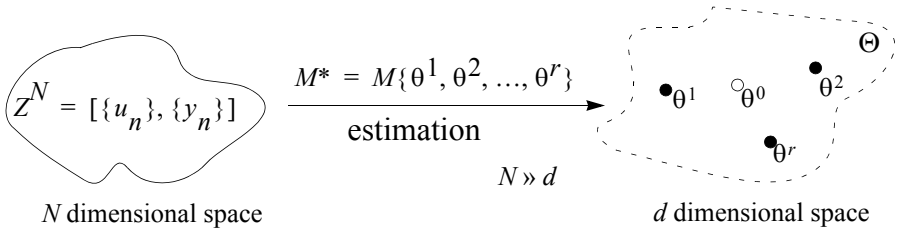


Figure 2-5 Model uncertainty set estimation

In many situations, an identified model can only be an approximation of the exact plant dynamics, due to the fact that the model set is restricted to models of limited order [42]. The requirement for the estimation procedure is that the estimate should belong to

the *parameter vector uncertainty set,* Θ, or in other words, to the *feasible parameter set* where $\Theta \in R^d$. The model uncertainty set is an estimated set of models that is used to represent the unavoidable incomplete knowledge of the unknown dynamic system to be controlled [21]. From the system identification's point of view, a model uncertainty set, Θ, consists of all models that are either validated or cannot be invalidated by the available data [21].

The source of uncertainty in the model originates from the so called parametric and nonparametric uncertainties. *Parametric uncertainty* refers to the possible error in the model parameters. *Nonparametric uncertainty* refers to the model error originating from unmodelled dynamics.

Any parameter vector that belongs to this uncertainty set, Θ, with the selected model structure, defines a particular model, $M(\theta^k)$, for the system. Using mathematical notations [30]

$$M \;:\; \Theta \ni \theta^k \rightarrow M(\theta^k). \tag{2.3}$$

The estimated parameter vector, θ^k, belongs to the uncertainty set, Θ, if it satisfies the bounds, γ, made by the requirements toward the estimation. In a formal way

$$\theta^k \in \Theta = \bigcap_{i=1}^{N} \{\theta^k : V_{n-i}(\theta^k_{n-i}, v_{n-i}) \le \gamma\} \tag{2.4}$$

where the function $V_n(\theta_n, v_n)$ describes the quantification of the selected requirements, n is the index of last observation, and N is the length of the observed data,

$Z^N = [\{u_n\}, \{y_n\}]$. A rational choice for a model uncertainty set, Θ, would be to collect all models that are not invalidated by the data and the prior information on the system, using (2.4) [48]. The selected model is not invalidated if it can account for all the previously observed input-output behavior within the tolerance limits. In section 2.4 validation of the model is considered through the concept of stability.

2.4 Practical Stability

In the instance of permanent closed-loop identification, the practical interest is on the behavior of the model over a fixed interval of time. The objective is to provide an accurate model which will describe the dynamic behavior of the system.

In the case of a "practical" system, it is quite certain that perturbations, captured as part of additive noise, will come up not only because of the initial conditions but also due to exterior actions. Will the selected model exhibit a response to possible disturbances within certain expected bounds? It appears reasonable to formulate such a question within a stability framework, i.e., a model is "stable" if it operates within the pre-specified bounds and is "unstable" if it does not. However, it is evident that the classical theory of stability requires strong modifications in order to be relevant toward the solution of such a stability question [49]. In the literature this problem is described as *practical stability* [50] or as *finite time stability* [49]. In this research, the idea of finite time stability is used as an inspiration for the model validation.

The main objective in any system identification is to minimize the prediction error and the parametric error under external disturbances.

- The *prediction error*, $v_n^k = y_n - \hat{y}_n^k$, for the k-th model is the difference between the plant output and the output predicted by the k-th model and it is used to characterize the error in modelling.
- The *parametric error*, $\omega_n^k = (\theta_n^k - \theta_n^0)$, is the difference between the "true system parameters" and the estimated ones by the k-th model and it is the error in the estimated parameters.

The objective function, which includes the prediction error as well as the parametric error, can be defined as

$$J_n = \min_k \sum_{n=1}^{N} (\theta_n^k - \theta_n^0)^T W_\theta (\theta_n^k - \theta_n^0) + v_n^k W_v v_n^k \tag{2.5}$$

where, $W_\theta \geq 0$ and $W_v \geq 0$ are freely selectable time varying weighting factors, $W_\theta \in R^{d \times d}$ and $W_v \in R^1$. The term under the summation can be expressed as a function of θ_n^k and v_n^k

$$V_n(\theta_n^k, v_n) = (\theta_n^k - \theta_n^0)^T W_\theta (\theta_n^k - \theta_n^0) + v_n^k W_v v_n^k . \tag{2.6}$$

This discrete function, $V_n(\theta_n^k, v_n^k)$, satisfies all the requirements of the *Lyapunov function* [50] in the discrete domain. Specifically, it is non-negative

$$V_n(\theta_n^k, v_n^k) \geq 0 \qquad \forall(\{\theta_n^k\}, \{v_n^k\}) \tag{2.7}$$

and it vanishes only at the origin $V_n(\theta_n^0, 0) = 0$, and $V_n(\theta_n^k, v_n^k) \to \infty$ as $\left\|\theta_n^k\right\| \to \infty$ or $\left\|v_n^k\right\| \to \infty$. The time difference is defined as

$$\Delta V_n(\theta_n^k, v_n^k) = \frac{[V_n(\theta_n^k, v_n^k) - V_{n-1}(\theta_{n-1}^k, v_{n-1}^k)]}{T_s} \tag{2.8}$$

where, T_s is the discrete-time step.

According to the Lyapunov stability theorem, a system is asymptotically stable if the time difference of its Lyapunov function is non-positive $\nabla V_n(\theta_n^k, v_n^k) \le 0$. The Lyapunov asymptotic stability requirement is satisfied if $\theta_n^k \to \theta_n^0$ and $v_n^k \to 0$ as $n \to \infty$. This stability definition is developed for an ideal system, which is not necessarily valid for real systems. The stability theory for "practical" systems, which is concerned with a physical system and environment, requires some modifications. It is quite certain that in a physical environment perturbations will occur not only as initial conditions, which is a base of the conventional Lyapunov stability theory, but also due to exterior forces. It is important to make it obvious that the system identification under the influence of perturbing forces is moving away from the optimal estimation, which will manifest in the increase of the Lyapunov function, $V_n(\theta_n^k, v_n^k)$, and its positive time derivative $\nabla V_n(\theta_n^k, v_n^k) > 0$. Contrary to the conventional stability theory, where the positive time derivative of the Lyapunov function immediately qualifies the system as unstable, in a practical stability environment the assumption is that the system can be regarded as stable even when it is losing its stability, if $V_n(\theta_n^k, v_n^k) < \gamma$, as depicted in Figure 2-6.

For the finite time environment, the quality measurements of stability should also be redefined. Since the word “asymptotic” has little meaning in the finite time contexts, the word “contractive” is used instead [49]. Using the idea of practical stability the question is, will a given system’s response to the exogenous perturbing forces, observed through its Lyapunov function over a fixed interval of time $T_d = N \cdot T_s$, be contained within certain specified bounds. A system is stable if it operates within pre-specified bounds and it is unstable if it does not [49].

A minimum requirement for a stable system identification is that the estimated parameters belong to the uncertainty set, $\theta_n^k \in \Theta$, as presented in Figure 2-5. This uncertainty set, $\theta_n^k \in \Theta$, can have an equivalent definition through a Lyapunov function, by introducing the corresponding bound, $V_n(\theta_n^k, v_n^k) \leq \gamma$. The system identification will be *contractively stable*, if under the perturbing forces the estimated parameters are close to the true system parameters, $\theta_n^k \approx \theta^0$, or an equivalent definition through the Lyapunov function, by introducing the corresponding bound, $V_n(\theta_n^k, v_n^k) \leq \alpha$. And further, an intermediate stability region, called *quasicontractive stability* region, $\alpha \leq V_n(\theta_n^k, v_n^k) \leq \beta$, can be introduced. Stability in finite time environment is illustrated with four trajectories of the $V_n(\theta_n^k, v_n^k)$ function, illustrated in Figure 2-6, where the trajectory of the Lyapunov function is observed for the time interval $t = \{T_n, \ldots, T_{n+N}\}$.

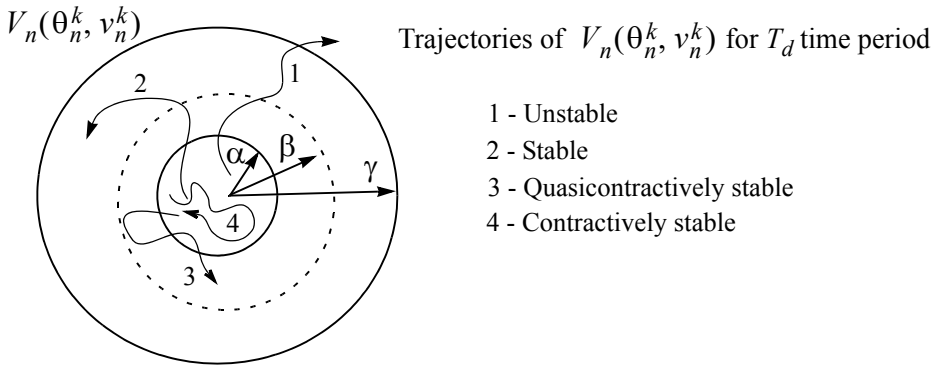


Figure 2-6 Definition of finite time stability

It is also important to point out that the perturbing forces should not be limited with any theoretically convenient limits, as it is suggested in most theoretical research, but instead they should be anticipated of any magnitude that is possible in the particular physical system. Under the assumption that the perturbing forces are not limited by convenient limits, the conventional stability analysis presented in numerous publications can not be implemented.

The stability of the model in this research is evaluated in real time. The argument in support of this method is that many investigations of stability that take account only of the model with linear approximations and introduce "proof by assumption," should not be taken too seriously [50-51]. This statement is very well sided with multiple model based identification, where the stability of the system identification is re-examined, based on observed information, in every T_d time period. If the model is found to be unstable or another method shows better performance, the used source of system parameter estimation shall be disqualified. A more stable source for the model as a source of the identified

parameters shall be selected from the competing system identification techniques. Evaluating the quality of the identified system model is not possible with the single system identification technique.

Based on the a priori information, the model selection process assumes a hypothesis that the selected model parameters will have the best ability to describe the dynamic behavior of the real system. The selected model parameters are valid for the following T_d time period, during which the model is reevaluated using the observed data from the plant. This process ensures that the system identification is stable based on the experimental information about the system, instead of the theoretical analysis, which will be defined in detail in the following sections.

2.5 Theory for Model Selection

In the multiple model estimation procedure, the implemented r algorithms calculate the model parameters for the selected model structure M. Although each algorithm results in an optimally estimated parameter vector, θ_n^k, these vectors differ from each other because of the different optimization criteria used to develop the particular algorithms. These estimated parameter vectors assemble the model set, M^*, at time $t = nT_s$, with r particular system models

$$M^* = \{\theta_n^1, \theta_n^2 \ldots \theta_n^r\}. \tag{2.9}$$

The task for the model selection is to select one parameter set, which will have the best chance to be the stable one in a sense of $V_n(\theta_n^k, v_n^k)$ functions in the next $T_d(\ = N \cdot T_s)$ intervals, where N is the number of collected discrete data points, $Z^N = [\{u_n\}, \{y_n\}]$. Practically, the *model selection process* assumes the hypothesis that the selected model will meet the performance specifications in the next T_d time periods based on the past observations. The ultimate objective of this selection process is to choose a model from the model set, M^*, and use that model to construct a model-based robust controller that best controls the plant.

In the literature [52], the previously described method of model validation which uses experimental data, is known as *model unfalsification*. Model unfalsification is based on the concept that the observed input-output data cannot be used to demonstrate the model's inability to provide the desired level of system model accuracy. In other words, the model selection process selects the best model from the model uncertainty sets. The optimal solution is to choose the model with the largest a posteriori probability. Specifically, the experimental data does not and can not guarantee that the model will meet the desired performance specifications in the future. At this point it should be clarified that the parameters for the uncertainty model set are estimated with different algorithms, each with guaranteed stability obtained in the process of particular algorithm development (H_2, H_∞, etc.). In the model selection process, instead of making the decision on theoretical bases, the unfalsification process should be used to validate the model. When the model is unfalsified (selected), it means that no evidence is available that contradicts the model stability [53].

Consequently, the optimal decision algorithm can be developed by reflecting on Bellman's principle of optimality, which states that "An optimal policy has the property that no matter what the previous decisions have been, the remaining decisions must constitute an optimal policy with regards to the state resulting from those previous decisions." [54]. According to Bellman, the optimal cost from time n ($t = nT_s$) equals to

$$J_n^*(\theta_n) = \min_{\theta_n^k \in M} \left(L_n(\theta_n, \phi_n) \right) + J_{n+1}^*(\theta_{n+1}) \tag{2.10}$$

where the optimal decision at time n is the θ_n^k that achieves this minimum. Equation (2.10) is the functional equation of dynamic programming, where $J_{n+1}^*(\theta_{n+1})$ represents the optimal cost from time $n+1$ to the terminal time, and $L_n(\theta_n, \phi_n)$ is the selected performance index.

In general implementation, the optimal control strategies must be determined by working backwards from the final stage. However, the solution of this equation for the model selection is simplified due to the fact that every decision point is independent from the previous and the following ones. Further, every decision that is made at time n is only valid for T_d time period, as presented in Figure 2-7, where the arrows represent the concurrently calculating algorithms for the parameter vectors, and the solid-line-arrows represent the algorithms with selected parameters.

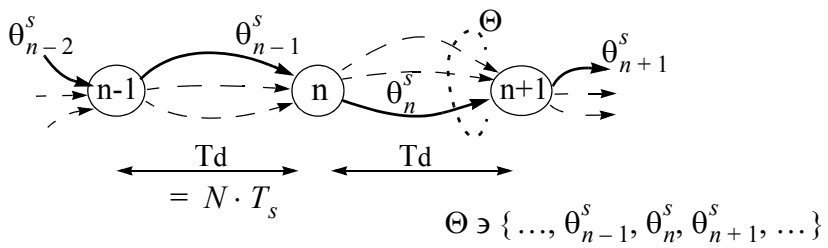


Figure 2-7 Decision points on the timeline

Taking into account the simplifications mentioned above, the solution of equation (2.10) results in the optimization equation

$$J^*_n(\theta_n) = \min_{\theta^k_n \in M} (L^k(\theta^k_n, \phi_n)) \rightarrow \theta^s_n \tag{2.11}$$

where $L^k(\theta^k_n, \phi_n)$ is the performance index defined for model validation and selection.

The *performance index* can be selected in many different ways as described in [55]. The goal is to predict the future behavior of the system dynamics, in the sense of model stability represented by function $V_n(\theta_n, v_n)$, using past observations. To accomplish this goal, the performance index $L^k(\theta^k_n, \phi_n)$ should be determined based on stability measurement function $V_n(\theta^k_n, v^k_n)$.

In order to better predict the system dynamics, the performance index requires further enhancements, using more observation data from the past. This idea is inspired by the way the human recollection or *reminiscence* works, using the memory of the past events and

experiences. Selection of the performance index, which entails the residuals and the mean square error of the estimated parameter vector, for model selection is detailed below.

2.6 Implementation of Model Selection

As stated above, the general theoretical background for model selection requires further specifications for practical implementation in a real time recursive environment. In the following this bridge between theory and practical implementation is given for the performance index (2.11) using the earlier definition of (2.6) and the theory which underpins it. First, the utilization of the prediction error is analyzed in a practical real time environment. This is followed by the analysis of the parametric error in a real time environment. And finally, the possible implementation of a reminiscence function as part of the performance index (2.11) is provided.

2.6.1 Prediction Error Calculation

The main performance index parameter is the residual v_n^k, the difference between the predicted and observed system output, which represents part of the output y_n that cannot be predicted by the model, from past data

$$v_n^k = y_n - \hat{y}_n^k(\theta_n^k) \tag{2.12}$$

where $\hat{y}_n^k(\theta_n^k)$ is the predicted system with the particular model $\hat{G}(\theta_n^k, T_s)$, as represented in Figure 2-8. The q^{-1} is the backward shift operator which introduces a sample delay in the observation sequence.

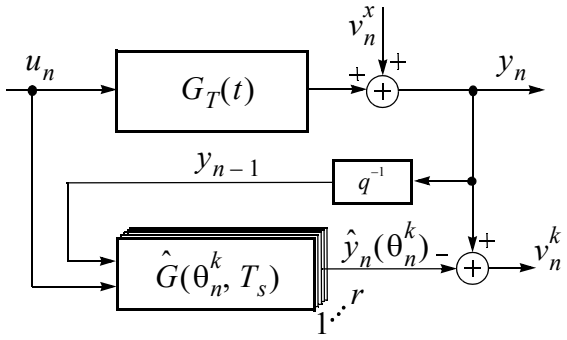


Figure 2-8 System identification

An important characteristic of a signal is its size. Norm functions are one way to determine the size of a signal. Collection of the residuals between two decisions constitutes a linear space of signals $l(R^N)$. The norms used to characterize this linear space are the l_p norms. The three commonly used l_p norms are:

the l_1 norm, which measures the sum of absolute values

$$l_1: \qquad \|e\|_1 = \sum_{i=1}^{N} |e_{n-i}|; \tag{2.13}$$

the l_2 norm, which measures the energy of the signal sequence

$$l_2: \qquad \|e\|_2 = \sum_{i=1}^{N} e_{n-i}^2; \text{ and} \tag{2.14}$$

the l_∞ norm, which measures the maximum value attained by the sequence

$$l_\infty: \qquad \|e\|_\infty = \max_{1 \le i \le N} |e_{n-i}|. \tag{2.15}$$

Both the l_1 and l_2 norms are measuring similar properties of the signal, $l_2 \le l_1 \le \sqrt{N}\, l_2$, but because the l_2 norm calculation is simpler, using it along with the l_∞ norm of the residuals is proposed for the model selection.

With the implementation of the above two norms, the selection is reduced to the estimators, which on the average yield the true parameter values. Hopefully, the estimators obtained in this way will produce values close to the true values most of the time [56].

2.6.2 Parametric Error Estimation

Due to the unknown true system parameters, θ_n^0, the direct calculation of the parametric error, $\omega_n^k = (\theta_n^k - \theta_n^0)$, as defined in (2.6) is not possible. Two alternative methods are presented below for evaluating the parametric error in system identification.

Approximation by the mean square error . One possible solution is to approximate the parametric error by the mean square error of the estimated parameter vector. This

approach is warranted if the goal is to find the estimator that exhibits the least variability in the parameter vector.

The mean square error (*mse*) of the estimated parameter vector can be used as the measure of variations in the estimated parameter vector

$$(\omega_n^k)^2 \approx mse(\theta_n^k) = E[(\theta_n^k - \theta_n^0)^T \cdot (\theta_n^k - \theta_n^0)]. \tag{2.16}$$

Due to the fact that the system's true parameter vector θ_n^0 is unknown, the mean square error calculation cannot be directly implemented. However, by introducing some approximations, a lower bound on the variations of the estimated parameter vector can be obtained. The mean square error on the $T_d = N \cdot T_s$ interval can be approximated as

$$\overline{mse}(\theta_n^k) = \frac{1}{N} \sum_{i=n-N}^{n} (\theta_n^k - \theta_n^0)^T \cdot (\theta_n^k - \theta_n^0). \tag{2.17}$$

The true parameter vector can be expressed as

$$\theta_n^0 = \overline{\theta_n^k} + \nabla^k; \qquad \overline{\theta_n^k} = \frac{1}{N} \sum_{i=n-N}^{n} \theta_i^k \tag{2.18}$$

where ∇^k is the bias in the averaged k-th estimated parameter vector $\overline{\theta_n^k}$. If (2.18) is substituted in (2.17), it is found that

$$\overline{mse}(\theta_n^k) = \sigma^2_{\theta_n^k} + (\nabla^k)^2 \tag{2.19}$$

the mean square error in the k-th estimated parameter vector is expressed through the calculated variance of the same

$$\sigma^2_{\theta_n^k} = \frac{1}{N} \sum_{i=n-N}^{n} (\theta_i^k - \overline{\theta_n^k})^T \cdot (\theta_i^k - \overline{\theta_n^k}) . \tag{2.20}$$

The introduced error is equivalent to the square of the bias. It is expected however, that all the implemented algorithms are unbiased, or in other words, the estimation is contractive with the bias error approaching zero as the estimators process more input-output information from the system.

Approximation by the prediction error . The assumption in this approach is that the model is linear, as defined in section 2.7. The derivation of approximation starts with the original formulations of the parametric error from (2.6)

$$(\theta_n^k - \theta_n^0)^T W_\theta (\theta_n^k - \theta_n^0) \tag{2.21}$$

where $W_\theta \geq 0$ is a freely selectable, time varying weighting factor, $W_\theta \in R^{d \times d}$. Without loosing any generality, the weighting factor can be selected as a function of the measured system input and output

$$W_\theta = \phi_n \phi_n^T \tag{2.22}$$

where ϕ_n is the measurement vector as defined in (2.32). If (2.22) is substituted in (2.21) the resulting equation is

$$(\theta_n^k - \theta_n^0)^T W_\theta (\theta_n^k - \theta_n^0) = (\theta_n^k - \theta_n^0)^T \phi_n \phi_n^T (\theta_n^k - \theta_n^0) \tag{2.23}$$

and by further manipulation it leads to

$$(\theta_n^k - \theta_n^0)^T W_\theta (\theta_n^k - \theta_n^0) = (\theta_n^{kT}\phi_n - \theta_n^{0T}\phi_n)(\phi_n^T\theta_n^k - \phi_n^T\theta_n^0) \tag{2.24}$$

where $\hat{y}_n^k(\theta_n^k)^T = \theta_n^{kT}\phi_n$, $y_n^T = \theta_n^{0T}\phi_n$, $\hat{y}_n^k(\theta_n^k) = \phi_n^T\theta_n^k$, and $y_n = \theta_n^{0T}\phi_n$ are the predicted and the measured system outputs. Further, the difference calculated is the prediction error (2.12) in the model

$$(\theta_n^k - \theta_n^0)^T W_\theta (\theta_n^k - \theta_n^0) = v_n^{kT} v_n^k . \tag{2.25}$$

In conclusion,

$$(\omega_n^k)^2 \approx v_n^{kT} v_n^k \tag{2.26}$$

the parametric error, $\omega_n^k = (\theta_n^k - \theta_n^0)$, in the performance index calculation as defined in (2.6) can be approximated by the prediction error square as shown in (2.14) by selecting a particular weighting factor which is the function of the measurement variables.

2.6.3 Decision Process Formulation

Formulation of the decision process for the expert system began with the analysis of the Lyapunov Stability theorem. As a next step, this theorem was combined with Bellman's principle of optimality, which led to the performance index $L^k(\theta, \phi)$ (2.11). Further, by extending this expression by the analysis of residuals and the mean square error from above, a new performance index has been derived for the k-th adaptive algorithm

$$L_n^k(\theta_n^k, \phi_n) = \kappa_2 \left\| e_n^k \right\|_2 + \kappa_\infty \left\| e_n^k \right\|_\infty + \kappa_\sigma \sigma^2_{\theta_n^k} + \kappa_\Sigma L_\Sigma^k \tag{2.27}$$

where κ_2 , κ_∞ , κ_σ, and κ_Σ are positive weighting coefficients. If the parametric error in (2.27) is approximated through the prediction error, the term $\kappa_\sigma \sigma^2_{\theta_n^k}$ is to be zero.

In order to have more control over the behavior of the performance index, a new term $\kappa_\Sigma L_\Sigma^{\ k}$ is included. The $L_\Sigma^{\ k}$ stands for the reminiscence, $R(x_n)$, of the k-th adaptive algorithm from the past

$$L_\Sigma^{\ k} = R\left(\left\| e_n^{\ k} \right\|_2 , \left\| e_n^{\ k} \right\|_\infty, \sigma^2_{\theta_n^k} \right) . \tag{2.28}$$

In the decision process the function of the reminiscence is based on the general observation that the model with a continuous good performance has the most chance to perform well in the next decision interval. The reminiscence function can be selected as

$$R(x_n) = \frac{1}{N_R} \cdot \sum_{i=0}^{N_R} x_{n-i} \cdot \lambda_R^{-i} \tag{2.29}$$

where x_n is the independent variable of the function $R(x_n)$, N_R is the number of the decision intervals included in reminiscence calculation, which corresponds to $t = T_d \cdot N_R$ time interval, and $0 \ll \lambda_R \leq 1$ is the reminiscent weighting factor. If the reminiscent weighting factor is selected as $\lambda_R = 1$, it has a uniform weighting on the data in the decision process. Accordingly, if it is selected as $\lambda_R < 1$, the recent performance of the estimation algorithm has more influence on the decision-making.

- The model, which shall best represent the system for the next T_d interval, is the model with the smallest performance index $L_n^{\,k}(\theta, \phi)$.

It is possible that during the adaptation period large perturbing forces may disturb the process resulting in disturbed estimated parameters. For this reason the last selected parameter vector, like a fixed model, will compete in every decision interval.

2.7 Linear System Model

In general situations, as in most technical systems, the relationship between the input sequence and the output sequence is nonlinear, which requires a nonlinear transfer function (2.2) for the model structure M. System identification of general nonlinear structures is quite demanding in the estimation mathematical procedures as well as in the observed data quality and quantity. The well practiced solution for this problem is the linearization of the nonlinear system around a certain trajectory, using the fact that the controller in closed-loop system identification can linearize the nonlinear system behavior on a relevant operating point. This trajectory will be a time varying function of the nonlinear system's operating point, with a linear model $G(\theta_n^0, T_s)$ using a time varying parameter vector θ_n^0.

The model-structure M for the linear system model is

$$G(\theta^0, T_s)u_n + H(\theta^0, T_s)\nu_n = \phi_n^T\theta_n + v_n. \tag{2.30}$$

The time-varying parameter vector is defined as

$$\theta_n = [d_{m_d}, a_1 \ldots a_{m_a}, b_1 \ldots b_{m_b}, c_1 \ldots c_{m_c}]^T \tag{2.31}$$

where m_d, m_a, m_b and m_c are the dimensions of the appropriate parameters, which are selected as zero if the corresponding parameter is not included in the model. The measurement vector is defined as

$$\begin{aligned} \phi_n = [1_{m_d}, -y_{n-1}, \ldots -y_{n-m_a}, u_{n-1}, \ldots u_{n-m_b}, \\ v_{n-1}, \ldots v_{n-m_c}]^T \end{aligned} . \tag{2.32}$$

The model formed by the product of $\phi_n^T \theta_n$ is the so called ARMAX model,

$$\hat{y}_n(\theta_n) = \phi_n^T \theta_n \tag{2.33}$$

consisting not only from the autoregressive (AR), $[a_1 \ldots a_{m_a}]^T$, and moving average (MA), $[b_1 \ldots b_{m_b}]^T$, but also from the noise model (X), $[c_1 \ldots c_{m_c}]^T$. The term represented by the $1_{m_d} \cdot d_{m_d}$, is the fourth component of the model, which is used to determine the constant (DC) part of the observed signal, $\{y_n\}$. If the observed signals contain a DC value the adaptation process will not work. This problem can be resolved by including in the model the term, $1_{m_d} = 1$, which allows the DC value to be estimated with d_{m_d}. Otherwise, if the observed signal $Z^N = [\{u_n\}, \{y_n\}]$ does not contain a DC value, the term can be excluded from the system identification.

In order to estimate the parameters of the linearized system model, which represents the actual nonlinear system at the time of system identification, persistent system identification is the only possible solution. However, this results in the problem of system identification in the closed-loop. In the closed-loop identification the output is being fed back to the input by the controller, where the system input is correlated with the system output, which can confuse the identification algorithm. The research literature on closed-loop identification is very extensive, for example see [21,30,40,52], suggesting that in the closed-loop environment, the *prediction error method* is one of the best approaches to be applied [31]. The literature on the algorithms using the prediction error method is extensive. In the next chapter some of these techniques are described that use the prediction error method for the system parameter estimation.

3 Parameter Estimation Strategies

There are many means available to provide an estimate for θ_n^0 using the observation of an N input-output data sequence, $Z^N = [\{u_n\}, \{y_n\}]$. The strategy for identification, T_i, can be defined as

$$\theta = T_i(\{u_n\}, \{y_n\}) \tag{3.1}$$

where T_i represents the selected strategy for identification. In mathematical terms

$$T_i : \quad Z^N \rightarrow \theta \in \Theta \tag{3.2}$$

the strategy of identification defines the way in which the information in Z^N is used to estimate the parameter vector θ^0. This can be interpreted as a problem of an overdetermined system [57], where there are more equations than unknowns,

$$A_n \cdot \theta_n = Y_n \tag{3.3}$$

the data matrix, $A_n \in \Re^{N \times (m_a + m_b + m_c)}$, and the observation vector, $Y_n \in \Re^N$, are given as

$$A_n = \begin{bmatrix} \phi_n \\ \phi_{n+1} \\ \cdots \\ \phi_{n+N} \end{bmatrix} \qquad \text{and} \qquad Y_n = \begin{bmatrix} y_n \\ y_{n+1} \\ \cdots \\ y_{n+N} \end{bmatrix} \tag{3.4}$$

where the measurement vectors are defined as

$$\phi_n = [1_{m_d} - y_{n-1}, \ldots - y_{n-m_a}, u_{n-1}, \ldots u_{n-m_b}, v_{n-1}, \ldots v_{n-m_c}], \tag{3.5}$$

and the parameter vector, to be calculated, is

$$\theta_n = [d_{m_d}, a_1 \ldots a_{m_a}, b_1 \ldots b_{m_b}, c_1 \ldots c_{m_c}]^T \tag{3.6}$$

where m_d, m_a, $m_b$and m_c are the dimensions of the appropriate measurements and parameters, which are selected as zero if the corresponding parameter is not included in the model.

Usually an overdetermined system has no exact solution $\theta \in \Re^{(m_a + m_b + m_c)}$, rather the solution can be found by selecting a strategy T_i

$$T_i : \quad \min_{\theta_n} \left\| A_n \cdot \theta_n - Y_n \right\|_p , \tag{3.7}$$

which minimizes the p norm of the prediction error. Selection of different norms renders different solution strategies T_i. For example if $p = 1$ it is the "Worst-Case Estimation" algorithm, if $p = 2$ it is the "Least Squares" algorithm, if $p = \infty$ it is the "Least Mean Squares" algorithm.

- The use of a $p = 1$ norm criterion is appropriate when it is suspected that a small portion of the data being analyzed is unreliable (i.e., contains data outliers). The $p = 1$ norm criterion has the capability of effectively ignoring a few bad data points while emphasizing the majority of data points which more properly reflects the true nature of the data [58].

- The use of a $p = 2$ norm criterion makes no assumption other than that the measurement errors, of the data being analyzed, have a normal distribution about the mean (i.e., broadband measurement noise) [59].
- The use of a $p = \infty$ norm criterion is appropriate when it is desired to find an approximate solutions to an inconsistent[1] system of linear equations T_i in which the largest error magnitude is to be minimized [58].

Many different parameter estimation techniques, which differ in the necessary a priori information, convergence speed, required calculation power, variances in estimated parameters, and robustness to external disturbances are described in the literature. Different situations make all techniques equally attractive. In order to select the right approach in every particular implementation, it would require the execution of countless tests with all available algorithms. To eliminate the long procedure of selection or the compromise between the techniques, an intelligent supervisor is proposed in section 2.5 “Theory for Model Selection” which in real-time continuously performs the evaluation of the implemented algorithms, and selects the identified parameters from the concurrently running algorithms. This approach not only simplifies the required research prior to the implementation, but also increases the robustness of the parameter identification without compromising the parameter identification convergence, speed, and robustness. This research focuses only on a few algorithms that were selected with the aim to achieve good performance, in terms of applicability, stability, accuracy, etc.

---

1. If there exists at least one choice for the vector θ_n which satisfies the linear relationship, $Y_n = A_n \cdot \theta_n$, then the system of equations is said to be consistent [58].

In the following sections, different parameter estimation algorithms, that are used in this research, are discussed.

3.1 H_2 Approach to System Identification

The so-called 'least squares' or H_2 technique is obtained if the strategy T_i for the estimation fits the model parameters $\hat{\theta}_n$ to the observed data by minimizing the quadratic norm, H_2, of the prediction error. The estimation strategy based on the observation of an N point input-output data sequence Z^N can be presented as [30]

$$T_i : \theta_n = \underset{\theta_i}{arg\ min} \left\{ \sum_{k=1}^{N} (y_k - \phi_k^T \theta_k)^2 \right\}. \tag{3.8}$$

The underlying assumption of the least squares technique is a priori knowledge of the statistical properties of the external perturbing forces. More specifically, it is the requirement towards the external disturbing forces to be stochastic, with Gaussian[1] distribution, in nature with known statistical properties [30].

There are a variety of choices for the estimation of system parameters θ_n under the H_2 criterion, yet the most widely used estimation algorithm is the Recursive Least Squares (RLS) algorithm, with a a forgetting factor λ. This forgetting factor is a parameter for

1. Karl Friedrich Gauss, 1777-1855, German mathematician

controlling the adaptation memory length. The use of a forgetting factor is intended to ensure that data points in the distant past are "forgotten" in order to afford the possibility of following the statistical variations of the observable data, when the filter operates in a non-stationary environment [47]. The relation between the forgetting factor and the "memory" length is given by the equation [60]

$$\lambda = e^{-1/l} \tag{3.9}$$

where l is the memory length measured in samples and e is the base of natural logarithm.

The update equation of the RLS algorithm is [47][30]

$$\theta_n = \theta_{n-1} + K_{RLS,n} \cdot v_n \tag{3.10}$$

$$v_n = y_n - \phi_n^T \cdot \theta_{n-1}. \tag{3.11}$$

The gain $K_{RLS,n}$ is calculated by

$$K_{RLS,n} = \lambda^{-1} \cdot P_{RLS,n-1} \cdot \phi_n \cdot R_{RLS,n}^{-1} \tag{3.12}$$

$$R_{RLS,n} = 1 + \lambda^{-1} \cdot \phi_n^T \cdot P_{RLS,n-1} \cdot \phi_n. \tag{3.13}$$

$P_{RLS,n}$ is the estimated error covariance of the system parameter vector

$$P_{RLS,n} = E[(\theta_n^0 - \theta_n) \cdot (\theta_n^0 - \theta_n)^T] \tag{3.14}$$

where θ_n^0 is the true system parameter vector, and θ_n is the estimated system parameter vector. $R_{RLS,n}$ is the estimated error variance of the predicted system output $\hat{y}_n = \phi_n^T \cdot \theta_n$, defined by the equation

$$R_{RLS,\,n} = E[(y_n - \hat{y}_n)^2] \tag{3.15}$$

where the estimated error covariance of the system parameter vector $P_{RLS,\,n}$, satisfies the Riccati recursion

$$\begin{aligned} P_{RLS,\,n} &= \lambda^{-1} \cdot P_{RLS,\,n-1} - \lambda^{-1} \cdot P_{RLS,\,n-1} \cdot \phi_n \cdot R_{RLS,\,n}^{-1} \cdot \phi_n^T \cdot P_{RLS,\,n-1} \\ &= \lambda^{-1} \cdot P_{RLS,\,n-1} - \lambda^{-1} \cdot K_{RLS,\,n} \cdot \phi_n^T \cdot P_{RLS,\,n-1} \end{aligned} . \tag{3.16}$$

The algorithm is initialized by setting $P_{RLS,\,0} = \delta^{-1} \cdot I$, where δ is a small positive constant and the parameter vector is set to $\theta_0 = 0$.

As described earlier, if the external disturbing forces satisfy the stochastic assumptions, the H_2-optimal estimators minimize the expected estimation error energy and yield to a maximum-likelihood estimate from the parameter vector θ_n. In practical applications, the assumption about stochastic disturbances cannot be always guaranteed. Therefore, the performance of the estimator in such an environment is undetermined or even un-robust where small deterministic disturbances can lead to large estimation error. The problem of robust estimation is a critical requirement in system identification. In the following section, an H_∞-optimal estimation algorithm will be used to address this question. In the H_∞-optimal framework, in order to achieve better robustness, the estimators minimize the maximum energy gain from the disturbances to the estimation error.

3.2 H_∞ Approach to System Identification

The approach that has been developed to address the estimation in the presence of deterministic disturbances is the so-called H_∞ estimation theory. The H_∞ estimation technique is obtained if the strategy T_i for the estimation fits the model parameters θ_n to the observed data by minimizing (or in the suboptimal case, bounds) the maximum energy gain from the external disturbing forces, which are unknown, but nonrandom

$$T_i : \theta_n = \underset{\theta_i}{arg\ sup} \underset{\nu_n}{} \left\{ \sum_{k=0}^{N} (y_k - \phi_k^T \theta_k) \right\}. \tag{3.17}$$

The goal in developing the H_∞ estimation algorithm is to minimize the worst-case energy gain. The resulting estimators will be robust with respect to disturbances and less sensitive to parameter variations, since no statistical assumptions are being made about the disturbances. Or in other words, the H_∞-optimal estimator guarantees the smallest estimation error energy over all possible disturbances of fixed energy. However, the resulting estimators may be over conservative.

Due to the minmax nature of the H_∞-optimal estimator, which means that the estimator minimizes the maximum energy of the disturbances, it can be treated as a game theoretical control problem with two players. Consequently, the nature (opponent) has access to the unknown disturbance sequences and chooses them to maximize the error energy gain in estimation. Concurrently, on the other side is the H_∞-optimal estimator, which

should be designed in such a way as to minimize this estimation error energy gain. Since the worst-case disturbance is a deterministic sequence, the H_∞-optimal estimator does not require any statistical assumptions on the disturbance signal [20].

The over conservative behavior of the H_∞ estimators often leads to slower convergence in the parameter estimation compared to the H_2 estimation technique. In contrast to H_∞, the H_2 optimal system parameter estimation offers faster system parameter estimation but with a less robust behavior in the case of external disturbances, especially if the disturbances violate the assumption that the external disturbance distribution is Gaussian. Despite the slower convergence of the H_∞ estimators, one of them is the least mean squares algorithm, which is widely used as an estimation technique. This algorithm has low computational requirements with excellent stability.

3.2.1 Least Mean Squares (LMS) Algorithms

Most of the estimation algorithms, such as the least squares, maximum likelihood estimation, require a priori knowledge of the statistical properties of the external disturbances. However, most of the applications (outside the research simulations) have none or very limited a priori information about these external disturbances. The introduction of the LMS adaptive filter by Widrow[1] and Hoff[2] in 1960 came as a significant development for a broad range of engineering applications, since the LMS algorithm does not require a pri-

1. Bernard Widrow 1929- , Stanford University, California
2. Marcian (Ted) Hoff 1937- , Rochester NY, Inventor of the microprocessor at Intel

ori statistical information. The only requirement is that the system and the model should be linear in the parameters.

Contrary to the original belief that the LMS algorithm is an approximate recursion for the least squares recursive problem, it should be regarded as the exact solution to an H_∞ optimization problem [20]. The LMS algorithm is a minmax algorithm. It minimizes the maximum energy gain from the disturbances to the prediction error. The closely related normalized LMS algorithm minimizes the maximum energy gain from the disturbances to the filtered errors. Moreover, both algorithms are also risk-sensitive optimal and tolerant to model uncertainties and parameter variations. The robust properties of the LMS and normalized LMS algorithms guarantee that the energy of the estimation error never exceeds the energy of the disturbances. It is shown in [20] that the LMS algorithm is the so-called central a priori H_∞-optimal algorithm, while the normalized LMS algorithm is a central a posteriori H_∞-optimal algorithm. These central filters possess the additional properties of being the maximum-entropy and risk-sensitive filters, as well as being the solution to certain quadratic dynamic games [20].[1]

The LMS and normalized LMS algorithms are gradient-based algorithms, where the estimates of the system parameters are updated along the negative derivation of the instantaneous estimates of the gradient vector [47].

The update equation of the LMS algorithm is [47][20]

$$\theta_n = \theta_{n-1} + \mu_{LMS} \cdot \phi_n \cdot v_n \tag{3.18}$$

1. See “All Stabilizing Controllers” on page 86.

$$v_n = y_n - \phi_n^T \cdot \theta_{n-1}. \tag{3.19}$$

The learning rate is selected according to

$$0 < \mu_{LMS} < \frac{1}{\left\| \phi_n^T \cdot \phi_n \right\|_\infty} \tag{3.20}$$

where the infinity norm calculates the maximum possible input power represented in the observed data. The presented bounds on the learning rate guarantees H_∞ optimality. If μ_{LMS} is selected accordingly to (3.20), then the LMS algorithm ensures that the energy of the predicted error will never exceed the energy of the disturbances. On the other hand, if (3.20) is not followed, the H_∞ optimality is no longer warranted, since it is a well-known fact that the LMS algorithm behaves poorly if the chosen learning rate is too large [20].

The update equation of the normalized LMS algorithm is [61]

$$\theta_n = \theta_{n-1} + \frac{\bar{\mu}_{LMS}}{(1 + \bar{\mu}_{LMS}) \cdot \phi_n^T \cdot \phi_n} \cdot \phi_n \cdot v_n \tag{3.21}$$

$$v_n = y_n - \phi_n^T \cdot \theta_{n-1}. \tag{3.22}$$

The normalized LMS algorithm converges in the mean square of prediction error, if the adaptation constant $\bar{\mu}_{LMS}$ satisfies [20] the following requirement

$$0 < \bar{\mu}_{LMS} < 1. \tag{3.23}$$

When a small value is assigned to $\bar{\mu}_{LMS}$, the adaptation is slow, which is equivalent to the LMS algorithm having a long “memory”. Correspondingly, the excess mean-squared error after the adaptation is small. On the other hand, when $\bar{\mu}_{LMS}$ is large, the adaptation is rel-

atively fast, but at the expense of an increase in the average excess mean-squared error after adaptation [47].

The normalized LMS algorithm exhibits a rate of convergence that is potentially faster than that of the standard LMS algorithm for both uncorrelated and correlated input data [20].

In conclusion, the general behavior of the H_2 and H_∞ estimators can be summarized in the following: The H_2-optimal estimator has the best average performance under the appropriate stochastic assumptions about the disturbances. However, it also has high estimation error peeks for some particular deterministic disturbances because of its un-robust characteristics. The H_∞-optimal estimator has the best worst-case performance but its average performance is much worse than that of the H_2 estimators.

3.3 Worst-Case Estimation

The basic idea underlying the "Worst-Case Estimation Theory" (WCE) is to assume magnitude bounds δ on the disturbances, and use these assumptions along with the observed input-output data Z^N in system identification. The method is eminently suitable for providing models completed with error bounds that are required for robust control design. This is so since the manifestation of unmodelled system dynamics can readily be formulated as a deterministic bounded disturbance [55]. The estimated model parameters θ_n are selected in a way that the inequality condition [55]

$$\left| y_n - \phi_n^T \theta_n \right| \leq \delta \tag{3.24}$$

must apply for every sample observed in an N point data record. In mathematical terms, this strategy can be formulated as

$$T_i \; : \; \theta_n \; = \; \underset{\theta_i}{arg} \left\{ \bigcap_{k=1}^{N} \{ \left| y_n - \phi_n^T \theta_n \right| \leq \delta \} \right\} \tag{3.25}$$

where the goal is to select a parameter vector θ_n, which will satisfy the inequality for all the observed data. The recursive solution to this strategy was first presented in [62]. The drawback of this approach is that in most cases the bound δ is not known a priory, but only as an approximate $\hat{\delta} > \delta$. If the approximate bound is significantly larger $\hat{\delta} \gg \delta$ than it is necessary, it will have degenerating effects on the estimated parameters. Next, using the approach published in [55], the theoretical background of the WCE theory will be presented.

The selected criterion in the WCE is that the residual of the model should be smaller than δ, which defines the limits for the parameter vector θ_n, presented in

$$y_n - \phi_n^T \cdot \theta \; = \; \pm\delta \; \rightarrow \; \; \theta \; = \; (\phi_n^T)^{-1} \cdot y_n \pm \delta \cdot (\phi_n^T)^{-1} \; . \tag{3.26}$$

The limits can be visualized as a pair of straight lines in parameter space that are orthogonal to ϕ_n. The calculation (3.25) then becomes the region bounded by the intersection of simple straight lines so that the parameter vector θ_n is in the convex polytope in the d

dimensional parameter space, where d is the dimension of θ_n^0. This is illustrated in Figure 3-1 where the feasible parameter set Θ is within the d dimensional parameter space.

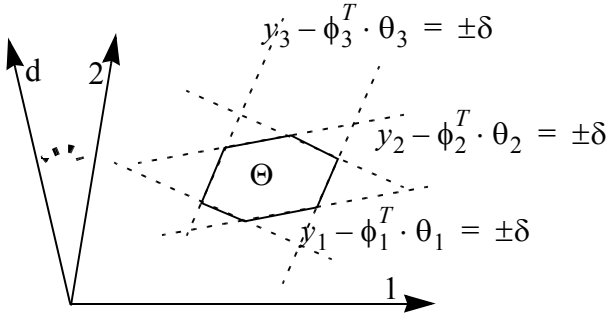


Figure 3-1 Feasible parameter set for the WCE

Direct implementation of this parameter estimation method, using linear programming with 2N constraints, is feasible but it is characterized with computationally intensive algorithms. An enhanced version of the WCE algorithm can be obtained if the absolute value calculation is replaced with a square, in which case the straight line bound region is replaced with an ellipsoidal overbounding region. For any choice of the positive weighting sequence $\{\rho_n\}$ the equation (3.25), satisfies the following inequality

$$\sum_{n=0}^{N-1} \frac{\rho_n}{\delta_n^2}(y_n - \phi_n^T\theta_n)^2 \le \sum_{n=0}^{N-1} \rho_n \qquad ;\ \rho_n > 0\,. \tag{3.27}$$

With an appropriate selection of the weighting sequence $\{\rho_n\}$ the estimation of θ_n can be optimized. A recursive solution for this was proposed in [62]. It was suggested that at each iteration the ellipsoid overbounding approximation should be made tight by mini-

mizing its volume, and that this can be achieved by choosing ρ_n as the largest positive root of the quadratic equation

$$(d-1)\cdot g_n^2\cdot\rho_n^2-((2d-1)\delta_n-g_n-v_n^2)\cdot g_n\cdot\rho_n+\delta_n(d(\delta_n-v_n^2)-g_n)\;=\;0 \qquad (3.28)$$

where $d = dim(\theta^0)$, $g_n = \phi_{n-1}^T\cdot P_{n-1}\cdot\phi_{n-1}$ and $v_n = y_n-\phi_n^T\cdot\hat{\theta}_{n-1}$. If no positive root exists, it means that the lines $y_n-\phi_n^T\cdot\hat{\theta}_{n-1}=\pm\delta_n$ do not intersect the old ellipsoidal region $(\theta^0-\theta_{n-1})^T\cdot P_{n-1}^{-1}\cdot(\theta^0-\theta_{n-1})$. In that case, no updating of regions should be done, which is accomplished by setting $\rho_n = 0$. Otherwise, the centre θ_n of the new ellipsoidal region becomes

$$\theta_n\;=\;\theta_{n-1}+\frac{P_{n-1}\cdot\phi_n\cdot v_n}{\delta_n/\rho_n+\phi_n^T\cdot P_{n-1}\cdot\phi_n} \qquad (3.29)$$

and the matrix P_n defines the size and orientation of the ellipsoid overbound

$$P_n\;=\;\beta_n\cdot\left(P_{n-1}-\frac{P_{n-1}\cdot\phi_n\cdot\phi_n^T\cdot P_{n-1}}{\delta_n/\rho_n+\phi_n^T\cdot P_{n-1}\cdot\phi_n}\right) \qquad (3.30)$$

with the gain term β_n calculated as

$$\beta_n\;=\;1+\rho_n-\frac{\rho_n\cdot v_n^2}{\delta_n^2+\rho_n\cdot\phi_n^T\cdot P_{n-1}\cdot\phi_n}\,. \qquad (3.31)$$

It is interesting to point out that the above described recursion depends non-linearly upon the observed input-output data, ϕ_n, because of the ρ_n term. The consequence for this is that the recursion can be repeated more than once on the same set of data (with re-

initialization $P_0 = P_n$ from the previous recursion) with the goal to find the smallest possible ellipsoidal bound on the feasible parameter set Θ.

By selecting for $\beta_n = 1$ the recursion for the worst-case parameter estimation is precisely that of the Kalman Filter estimation algorithm where the sequence of residuals, $\{v_n\}$, is a white Gaussian noise with variance δ_n^2 / ρ_n [55].

3.4 Kalman Filter Based System Identification

The Kalman Filter with its different derivatives is perhaps the most widely known algorithm for recursively estimating dynamic state vectors x_n that evolve according to the difference equations known as state-space model [20] [63-64]

$$\begin{aligned} x_{n+1} &= A_n \cdot x_n + B_n \cdot u_n + G_n \cdot w_n \\ y_n &= C_n \cdot x_n + D_n \cdot u_n + v_n \\ s_n &= L_n \cdot x_n + M_n \cdot u_n \end{aligned} \tag{3.32}$$

where x_n is the d dimensional state-space variable $x_n \in R^d$, u_n is the control input $u_n \in R^m$, w_n is the exogenous input which may be interpreted as a process-noise or driving disturbance $w_n \in R^r$, s_n is the desired system output $s_n \in R^q$, y_n is the measured output $y_n \in R^p$, v_n is an additive disturbance which can be regarded as a measurement noise $v_n \in R^p$. The corresponding system matrices, possibly time varying, have the following

dimensions: $A_n \in R^{d \times d}$, $B_n \in R^{d \times m}$, $G_n \in R^{d \times r}$, $\gamma_n \in R^{p \times}$, $D_n \in R^{p \times m}$, $L_n \in R^{q \times n}$, and $M_n \in R^{q \times m}$.

The Kalman Filter derivation is based on the least squares estimation strategy with assumptions about certain quantities that define the stochastic identification environment [65]. These values are presumed to be known. They include the covariance matrices for measurement-noise R_n, and the process-noise Q_n, assuming that the mean values are zero or known constants. The covariance matrices are given by the statistical expectation operation

$$E\left[\begin{bmatrix} x_0 \\ w_n \\ v_n \end{bmatrix}\begin{bmatrix} x_0 & w_j & v_j \end{bmatrix}\right] = \begin{bmatrix} \Pi_0 & 0 & 0 \\ 0 & Q_n\delta_{n,j} & 0 \\ 0 & 0 & R_n\delta_{n,j} \end{bmatrix} \tag{3.33}$$

where x_0 is the initial state disturbance, w_n is the process-noise and v_n is the measurement noise, and $\delta_{n,j}$ is the delta operator ($\delta_{n,j} = 0$ if $j \neq n$). The degree to which this information is correct is critical for the estimation performance and stability. Normally, this information is not readily available. The correct values can be estimated separately or by using the trial and error method. In mathematical terms, the Kalman Filer can be described as a minimum mean-square error optimization

$$T_i : \theta_n = \underset{\theta_i}{arg\ min}\left\{\sum_{k=0}^{N}(y_k - \phi_k^T\theta_k)^2 \middle| R_n, Q_n\right\}. \tag{3.34}$$

The solution to this problem is given by the Kalman Filter recursion [20]. In general form the desired system output is presented as

$$\hat{s}_n = L_n \hat{x}_n \tag{3.35}$$

where $\hat{x}_n$ satisfies the filtered form of the Kalman Filter recursion

$$\hat{x}_n = \hat{x}_n^- + K_{KF,n} \cdot (y_n - C_n \cdot \hat{x}_n^-) \tag{3.36}$$

$$\hat{x}_n^- = A_n \cdot \hat{x}_{n-1} + B_n u_{n-1} \tag{3.37}$$

$$K_{KF,n} = P_{KF,n-1} \cdot C_n^T \cdot R_{KF,n}^{-1} \qquad R_{KF,n} = R_n + C_n \cdot P_{KF,n-1} \cdot C_n^T \tag{3.38}$$

and $P_{KF,n}$ satisfies the Riccati recursion

$$\begin{aligned} P_{KF,n} &= A_n \cdot P_{KF,n-1} \cdot A_n^T + G_n \cdot Q_n \cdot G_n^T \\ &- A_n \cdot P_{KF,n-1} \cdot C_n^T \cdot R_{KF,n}^{-1} \cdot C_n \cdot P_{KF,n-1} \cdot A_n^T \end{aligned} . \tag{3.39}$$

When Kalman Filter is used for system parameter estimation, the recursive equation is in the state-space form. The system model $y_n = \phi_n^T \cdot \theta_n + v_n$ can easily be rewritten in the state-space form by selecting

$$A_n = I, B_n = 0, G_n = I, C_n = \phi_n, D_n = 0, L_n = I, M_n = 0 \text{, and, } x_n \equiv \theta_n . \tag{3.40}$$

In this way equation (3.32) becomes

$$\begin{aligned} x_{n+1} &= A_n \cdot x_n + G_n \cdot w_n = x_n + w_n \\ y_n &= C_n \cdot x_n + v_n \\ s_n &= L_n \cdot x_n \end{aligned} \tag{3.41}$$

where $x_n = \theta_n$ is the system parameter vector to be determined, w_n is the system noise the residual between two estimations, $C_n = \phi_n$ is the regression vector containing the observed data, v_n is the measurement noise which corrupts the observed system output, and s_n is the desired system output.

It is interesting to point out that the desired system output s_n for the selected $L_n = I$ is the estimated system parameter θ_n. On the other hand, if for $L_{\mathrm{n}} = \phi_n$ would be selected the desired output will be the filtered (uncorrupted) output of the system.

By applying the introduced transformations (3.41) the Kalman Filter based system parameter estimation can be expressed in the following form

$$\theta_n = \bar{\theta}_n + K_{KF,n} \cdot (y_n - \phi_n \cdot \bar{\theta}_n) \tag{3.42}$$

$$\bar{\theta}_n = \theta_{n-1} \tag{3.43}$$

where the Kalman gain $K_{KF,n}$ is calculated by

$$K_{KF,n} = P_{KF,n-1} \cdot \phi_n \cdot R_{KF,n}^{-1} \qquad R_{KF,n} = R_n + \phi_n^T \cdot P_{KF,n-1} \cdot \phi_n \tag{3.44}$$

and $P_{KF,n}$ satisfies the Riccati recursion,

$$P_{KF,n} = P_{KF,n-1} + Q_n - P_{KF,n-1} \cdot \phi_n \cdot R_{KF,n}^{-1} \cdot \phi_k^T \cdot P_{KF,n-1}. \tag{3.45}$$

In the above equations, $P_{KF,n}$ is the estimated error covariance of the system parameter vector

$$P_{KF,n} = E[(\theta_n^0 - \theta_n) \cdot (\theta_n^0 - \theta_n)^T] \tag{3.46}$$

where θ_n^0 is the true system parameter vector and θ_n is the estimated system parameter vector. The estimated error variance of the predicted system output $\hat{y}_n = \phi_n \cdot \theta_n$ is calculated by the recursion for $R_{KF,n}$, and it is defined as

$$R_{KF,n} = E[(y_n - \hat{y}_n)^2]. \tag{3.47}$$

In the Kalman Filter equations the covariance matrices for measurement-noise R_n, and the process-noise Q_n represent a priori knowledge of the external disturbing forces. At the same time, these parameters can be used to control the estimation process by the Kalman Filter, such as the speed of the estimation versus variance in the estimated parameters. The correlation matrices, R_n and Q_n can be defined using a parameter for each, to be placed along the matrix diagonals. This will ensure that these matrices are symmetric. The filter algorithm obtained in this way is also known as the random-walk Kalman Filter. These values can be estimated using the residuals in the system output and the variances between two estimated system parameters

$$R_n \sim E[(y_n - \hat{y}_n)^2] \tag{3.48}$$

$$Q_n \sim E[(\theta_n - \theta_{n-1})^2]. \tag{3.49}$$

3.5 Array Algorithms for System Identification

Since the first array form of the Kalman Filter was developed in the 1960s, an intensive research in this field has taken place resulting in various solutions for the array algo-

rithms. Basically, the least square array algorithms for adaptive filtering can be divided into two big families: the *transversal adaptive filtering algorithms;* and for the order-recursive *lattice adaptive filtering algorithms.* Both are characterized with an order of magnitude reduction in computation effort, compared to the RLS algorithm. The main characteristics of these algorithms are the reduced need in numerical range for the calculation (well-conditioned) and the use of orthogonal transformations, which typically leads to more stable algorithms.

In compact form, array algorithms can be described via rotation operations on a pre-array of numbers, which have to be triangularized in order to yield a post-array of numbers. The desired quantities can be read out from the post-array. In schematic way, this can be presented as [66]

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} \cdot \Theta_R = \begin{bmatrix} x & 0 & 0 & 0 \\ x & x & 0 & 0 \\ x & x & x & 0 \\ x & x & x & x \end{bmatrix} \tag{3.50}$$

where Θ_R is any rotation matrix that triangularizes the pre-array. The different realization of the triangularization [57] and its algorithmic realization lead to a number of "different" algorithms.

In this research two least squares or H_2 transversal filtering algorithms are selected for implementation: the conventional QR algorithm (QR-RLS) and the Inverse QR algorithm (IQR-RLS) algorithm. The LMS algorithm also has a QR based realization (QR-LMS) [66] which is also included.

3.5.1 QR-RLS Algorithm

Recursion for the QR-RLS algorithm can be summarized as a triangularization of the pre-array, where ϕ_n is the observation vector $\phi_n \in R^{d \times 1}$ updated in every recursion, y_n is the observed system output $y_n \in R$, λ $(0 \ll \lambda \leq 1)$ is the forgetting factor. The internal variables $p_{n-1} \in R^{d \times 1}$ and the lower triangular correlation matrix $\Phi_{n-1}^{1/2} \in R^{d \times d}$, initialize as $p_0 = o$ and $\Phi_0^{1/2} = \delta^{1/2} I$, where δ is a small positive constant and o is a zero vector [66][67]

$$\begin{bmatrix} \lambda^{1/2} \cdot \Phi_{n-1}^{1/2} & \phi_n \\ \lambda^{1/2} \cdot p_{n-1}^T & y_n \\ o^T & 1 \end{bmatrix} \cdot \Theta_{R,n} = \begin{bmatrix} \Phi_n^{1/2} & o \\ p_n^T & v_n \cdot \gamma_n^{1/2} \\ \phi_n^T \cdot \Phi_n^{-T/2} & \gamma_n^{1/2} \end{bmatrix}. \tag{3.51}$$

The estimation of the system parameters can be obtained by solving the upper triangular system of the linear equations from the post-array

$$\theta_n^T \cdot \Phi_n^{1/2} = p_n^T. \tag{3.52}$$

The computation complexity of this algorithm is $O(d^2)$ per recursion.

3.5.2 IQR-RLS Algorithm

Recursion for the IQR-RLS algorithm can be described as a triangularization of the pre-array, where ϕ_n is the observation vector $\phi_n \in R^{d \times 1}$ updated in every recursion, y_n is

the observed system output $y_n \in R$, λ $(0 \ll \lambda \leq 1)$ is the forgetting factor. The internal variables $p_{n-1} \in R^{d \times 1}$ and the lower triangular correlation matrix $\Phi_{n-1}^{1/2} \in R^{d \times d}$, initialize as $p_0 = o$ and $\Phi_0^{1/2} = \delta^{1/2} I$, where δ is a small positive constant and o is a zero vector [66][68]

$$\begin{bmatrix} 1 & \lambda^{-1/2} \cdot \phi_n^T \cdot \Phi_{n-1}^{-1/2} \\ o & \lambda^{-1/2} \cdot \Phi_{n-1}^{-1/2} \end{bmatrix} \cdot \Theta_{R,k} = \begin{bmatrix} \gamma_n^{-1/2} & o^T \\ k_n \cdot \gamma_n^{-1/2} & \Phi_n^{-1/2} \end{bmatrix}. \tag{3.53}$$

The estimation of the system parameters is calculated by the equation

$$\theta_n = \theta_{n-1} + [k_n \cdot \gamma_n^{-1/2}] \cdot [\gamma_n^{-1/2}]^{-1} \cdot (y_n - \phi_{n-1}^T \cdot \theta_{n-1}) \tag{3.54}$$

where the quantities $(k_n \cdot \gamma_n^{-1/2})$ and $\gamma_n^{-1/2}$ are read directly from the first and from the last line in the post-array. The computation complexity for this algorithm is $O(d^2)$ per recursion.

3.5.3 QR-LMS Algorithm

The very popular H_∞ norm based LMS algorithm also has QR decomposition based realization, which shows better numerical properties than the original one [69]. Following the derivation in [69], the starting point is the well known LMS stochastic gradient algorithm. The goal is to transform the LMS algorithm into the normal equation form

$\theta_n = R^{-1}b$, which is the base for the QR algorithm development. A different derivation can be found in [20]. The equations for the LMS algorithm are

$$\theta_n = \theta_{n-1} + \mu \cdot \phi_n \cdot v_n \tag{3.55}$$

$$v_n = y_n - \phi_n^T \cdot \theta_{n-1} \tag{3.56}$$

where μ is the step size parameter, which satisfies the H_∞ norm requirement

$$\mu = \frac{1}{a^2 + \phi_n^T \cdot \phi_n} \tag{3.57}$$

$a > 0$ is the weighting factor for the step size. Inserting (3.56) and (3.57) into (3.55)

$$\theta_n = \theta_{n-1} + \frac{\phi_n \cdot (y_n - \phi_n^T \cdot \theta_{n-1})}{a^2 + \phi_n^T \cdot \phi_n} \tag{3.58}$$

and expanding this equation with $\pm(a^{-2} \cdot y_n \cdot \phi_n \cdot \phi_n^T \cdot \phi_n)$

$$\theta_n = \theta_{n-1} + \frac{y_n \cdot \phi_n - \phi_n \cdot \phi_n^T \cdot \theta_{n-1} + a^{-2} \cdot y_n \cdot \phi_n \cdot \phi_n^T \cdot \phi_n - a^{-2} \cdot y_n \cdot \phi_n \cdot \phi_n^T \cdot \phi_n}{a^2 + \phi_n^T \cdot \phi_n} \tag{3.59}$$

and by using the identity

$$y_n \cdot \phi_n + a^{-2} \cdot y_n \cdot \phi_n \cdot \phi_n^T \cdot \phi_n = (a^{-2} \cdot y_n \cdot \phi_n) \cdot (a^2 + \phi_n^T \cdot \phi_n) \tag{3.60}$$

the equation can be further rearranged as

$$\theta_n = \theta_{n-1} - \frac{\phi_n \cdot \phi_n^T \cdot \theta_{n-1}}{a^2 + \phi_n^T \cdot \phi_n} + a^{-2} \cdot y_n \cdot \phi_n - \frac{a^{-2} \cdot y_n \cdot \phi_n \cdot \phi_n^T \cdot \phi_n}{a^2 + \phi_n^T \cdot \phi_n} \tag{3.61}$$

$$\theta_n = \left(I - \frac{\phi_n \cdot \phi_n^T}{a^2 + \phi_n^T \cdot \phi_n}\right) \cdot \theta_{n-1} + \left(I - \frac{\phi_n \cdot \phi_n^T}{a^2 + \phi_n^T \cdot \phi_n}\right) \cdot a^{-2} \cdot y_n \cdot \phi_n \tag{3.62}$$

$$\theta_n = a^{-2}\left(I - \frac{\phi_n \cdot \phi_n^T}{a^2 + \phi_n^T \cdot \phi_n}\right) \cdot (a^2 \cdot \theta_{n-1} + y_n \cdot \phi_n). \tag{3.63}$$

Using the matrix inversion lemma [47]

$$\begin{aligned} A &= B^{-1} + C \cdot D^{-1} \cdot C^T \\ A^{-1} &= B - B \cdot C \cdot (D + C^T \cdot B \cdot C)^{-1} \cdot C^T \cdot B \end{aligned} \tag{3.64}$$

$$B^{-1} = a^2 \cdot I \qquad D = I \qquad C = \phi \tag{3.65}$$

the equation can be transformed into

$$a^{-2} I - \frac{a^{-2} I \cdot \phi_n \cdot \phi_n^T \cdot a^{-2} I}{(a^2 + \phi_n^T \cdot \phi_n) \cdot a^{-2} I} = (a^2 \cdot I + \phi_n \cdot \phi_n^T)^{-1}. \tag{3.66}$$

Further rearrangement results in

$$\begin{aligned} \theta_n &= (a^2 \cdot I + \phi_n \cdot \phi_n^T)^{-1} \cdot (a^2 \cdot \theta_{n-1} + y_n \cdot \phi_n) \\ &= \left\{ \begin{bmatrix} aI & \phi_n \end{bmatrix} \cdot \begin{bmatrix} aI \\ \phi_n^T \end{bmatrix} \right\}^{-1} \cdot \left(\begin{bmatrix} aI & \phi_n \end{bmatrix} \cdot \begin{bmatrix} a\theta_{n-1} \\ y_n \end{bmatrix} \right) \cdot \end{aligned} \tag{3.67}$$

This representation of the conventional LMS equation has the well known normal equation form $\theta_n = R^{-1} \cdot b$ of the estimation process. The array algorithm form can be acquired when the matrix inverse calculation is performed using the QR decomposition

algorithm. The above described solution has improved numerical properties but it requires more calculations compared to the conventional LMS algorithm.

Part - II Robust Control

Robust control theory has emerged in response to the need to design controllers for complex dynamic systems, where the classical control system design techniques, with their trial and error approach, are unsatisfactory. The development of robust control theory has arrived from the realization that the model representing the plant has some level of uncertainty. The uncertainty arises in the mathematical models that are only approximate representations of the real plant and it's exogenous signals. Most controllers implemented in industrial environments, where the real issue is the difficulty of modelling the plant to be controlled, belong to this category. Historically, one can think of the problem of robust stabilization as having its origin in the conjectures of Aizerman [70] and Kalman [71], i.e. long before the word 'robustness' itself come into vogue [72]. In the early 1980s, with considerable mathematical basis, Zames [8] defined the fundamentals of robust control with the original formulation of the H_∞ control theory. In the following decade, the time domain formulation evolved [73], built on the game theory, which resulted in a Riccati equations based solution. Since the early 1990s the discrete-time solution has been defined in research papers [74-77] and published in a number of books [78-85].

In this Part, a brief introduction to H_∞ control theory is provided in the context of the noncooperative dynamic game theory. The practical aspects and a numerical implementation of the control design will also be demonstrated.

4 Preliminary to Robust Control

Robust control is a term used to define a control system design where, beside the requirements for stability, a desire to minimize the preliminary assumptions about the plant and the disturbances effecting it are also present. The robust control investigated in this research is the “worst-case design”. In this approach, the preliminary assumption is that the disturbances affecting the plant have the worst possible effect from the control point of view, however these effects are limited by their intensity. Further research encompassing these ideas resulted in the development of the H_∞ (sub)optimal control algorithm. The implementation of this algorithm is also investigated.

Some underlying theories that are essential to understand and to successfully implement H_∞ (sub)optimal control in discrete time are introduced in this thesis. First, the plant model used for the control system design is examined. It is reflected upon the fact that, in most of the realistic applications, the exact plant model is unknown. This lack of knowledge brings uncertainty to the design. To address and characterize this uncertainty by the control system, the *small gain theorem* is introduced. Finally, some properties of the H_∞ (sub)optimal control are described.

4.1 Uncertainty in Plant Model

For the control system design a discrete-time mathematical model, $\hat{G}(\theta, T_s)$, of the plant is required. The purpose of this model, $\hat{G}(\theta, T_s)$, is to map the control inputs to the outputs. Due to the fact that only the finite time and disturbed observations from the plant are available, it is unlikely that even the most complex mathematical model can represent the true physical plant. On the other hand, it is usually advantageous for the implementation if the mathematical representation is a simpler one, concentrating only on the behavior that is relevant for control. To be practical, a design technique must account for the inevitable inadequacy of models. A good model should be simple enough to facilitate the design, yet complex enough to give the engineer confidence that designs based on the model will work on the true plant [83].

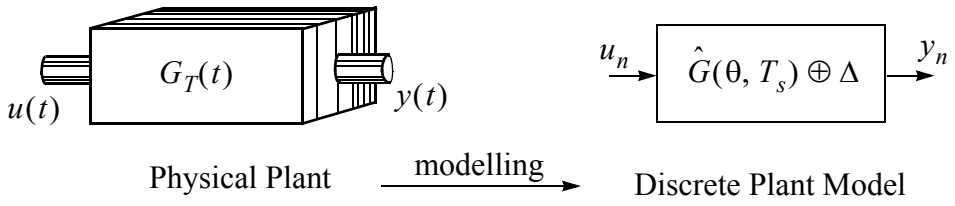


Figure 4-1 Plant modelling

Figure 4-1 illustrates the modelling of a physical plant, $G_{T_s}(t)$, with a discrete plant model, $\hat{G}(\theta, T_s)$. The difference between the observed plant, $G_{T_s}(t)$, and the representa-

tion of the model, $\hat{G}(\theta, T_s)$, is the *model uncertainty,* Δ, [83]. To achieve a high level of performance in the design of the control system, this model uncertainty must be taken into account.

The model uncertainty arises from the fact that the plant model, $\hat{G}(\theta, T_s)$, is only an approximation of the true model, $\hat{G}(\theta + \theta_\nabla, T_s)$, where θ_∇ is an unknown but bounded, random variable. If the bound on the variable θ_∇ can be estimated in some way as $\pm\theta_{\nabla MAX}$, the uncertain model would be defined by the model and the bound $\hat{G}(\theta + \theta_{\nabla MAX}, T_s)$. Uncertainty models inherently describe both a nominal model and a bound on the model accuracy [86]. In other words, an uncertainty model is simply a description of a set of possible models for an unknown plant.

4.2 Linear Fractional Transformation

A widely accepted approach to the *representation of uncertainty,* Δ, is the so called *Linear Fractional Transformation* (LFT) uncertainty model [7][80], as shown in Figure 4-2. The LFT model enables the known part $\hat{G}(\theta, T_s)$ to be separated from the unknown part Δ of the plant model.

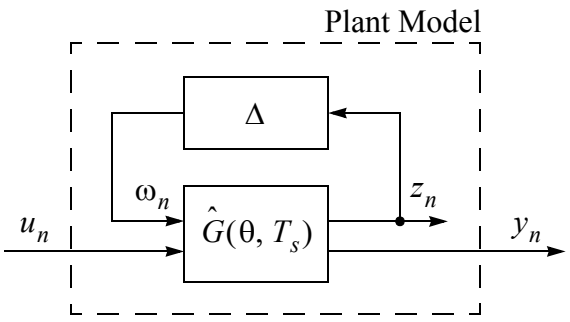


Figure 4-2 Plant model with uncertainty

The plant model, $\hat{G}(\theta, T_s)$, is extended with a new set of inputs ω_n, which interpret the effects of the model uncertainty to the behavior of the model. The signal ω_n, as a time function belongs to the l^2 space (square sumable), and can be considered as the *process-noise*, which is generated by the model uncertainty system Δ. Furthermore, Δ interprets all the possible exogenous disturbances acting on the plant. The set of control signals u_n is used to influence the dynamic behavior of the plant. This control action, based on the present and past observations of the system outputs y_n, is calculated by the controller Σ_K. The additional output quantity z_n, the controlled output of $\hat{G}(\theta, T_s)$, is a performance measure, depending on the particular problem at hand [84]. The goal of the controller Σ_K is to stabilize the plant under the acts of exogenous disturbances.

4.3 Small Gain Theorem

The plant stability affected by the uncertainty in the model, as depicted in Figure 4-2, can be assessed using the *small gain theorem* [7][80]. This is shown in more detail in Figure 5-3. The small gain theorem is based on the fact that a plant modeled by the closed-loop $z_n \rightarrow \omega_n \rightarrow z_{n+1}$ of $\hat{G}(\theta, T_s)$ and Δ, will be stable if the closed-loop gain is less than one.

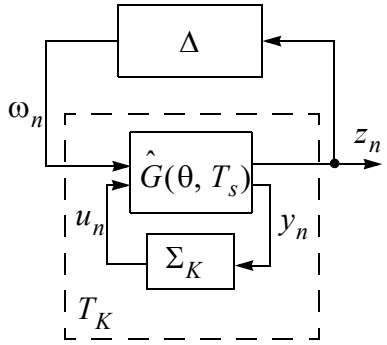


Figure 4-3 Small gain theorem model

This theorem can be more precisely interpreted using the *four blocks technique*. The LFT plant model, $\hat{G}(\theta, T_s)$, in Figure 4-3, has two inputs and two outputs. Therefore, it can be partitioned into a four block transfer matrix form, where the output z_n is a function of both inputs ω_n and u_n. The control signal, u_n, is calculated by the controller

$$\Sigma_K : \quad u_n = K \cdot y_n \tag{4.1}$$

from the model's other output, y_n. In mathematical terms, this can be expressed as

$$\begin{aligned} z_n &= \hat{G}_{11} \cdot \omega_n + \hat{G}_{12} \cdot u_n \\ y_n &= \hat{G}_{21} \cdot \omega_n + \hat{G}_{22} \cdot u_n \\ u_n &= K \cdot y_n \end{aligned} \qquad where \qquad \begin{bmatrix} \hat{G}_{11} & \hat{G}_{12} \\ \hat{G}_{21} & \hat{G}_{22} \end{bmatrix} = \hat{G}(\theta, T_s) \tag{4.2}$$

where $\omega_n \in R^m$, $u_n \in R^l$, $z_n \in R^q$, $y_n \in R^p$ and the transfer matrices have the appropriate dimensions. The transfer function from $\omega_n \to z_n : z_n = T_K \cdot \omega_n$ can be obtained as

$$T_K = \hat{G}_{11} + \hat{G}_{12} \cdot (I - K \cdot \hat{G}_{22})^{-1} \cdot K \cdot \hat{G}_{21} \tag{4.3}$$

where the transfer function, T_K, represents the plant with the closed-loop control system.

As mentioned earlier, if ω_n is a time function in l^2 space, and if the transfer function T_K has a finite gain, then the output z_n will also belong to l^2 space. The *small gain theorem* requires the worst-case closed-loop gain to be less than one, or

$$\|\Delta\|_\infty \cdot \|T_K\|_\infty < 1 \quad . \tag{4.4}$$

Moreover, it is required that the uncertainty, Δ, be small in the sense that

$$\|\Delta\|_\infty < \frac{1}{\gamma} . \tag{4.5}$$

Then, for the plant stability it is required that the maximum transfer gain from ω_n to z_n be less than γ

$$\|T_K\|_\infty < \gamma . \tag{4.6}$$

Equation (4.6) is interpreted as a necessary condition to implement a robust control, if the model uncertainty is bounded by the equation (4.5). Consequently, an internally stabi-

lizable controller Σ_K can be formulated based on the knowledge that the uncertainty is bounded by $1/\gamma$, such that the transfer function is $\|T_K\|_\infty < \gamma$, if it exists. From equation (4.6), γ is seen as an *attenuation factor,* which is a function of the controller Σ_K. The lower limit on the minimum achievable value for γ is represented with γ^* and is called *infimum*. This can be regarded as the fundamental idea for the H_∞ (sub)optimal control design.

The H_∞ norm of the transfer function from $\omega_n \rightarrow z_n : z_n = T_K \cdot \omega_n$ is defined as the maximum energy gain from the disturbances ω_n on the regulated signal z_n [75] [76]

$$\|T_K\|_\infty = \sup_{\omega \in l_2,\, \omega \neq 0} \frac{\|z\|_2}{\|\omega\|_2}. \tag{4.7}$$

In other words, the H_∞ norm is defined as a norm on transfer matrices, when translated to the time domain, the H_∞ norm is nothing else than the l_2-*induced norm* (from the input time-function to the output time-function for initial state zero) [87]. The goal in H_∞ (sub)optimal control design is to minimize the worst-case energy gain, and the claim is that the resulting controllers Σ_K will be robust with respect to disturbance variations, since no statistical assumptions are being made about the disturbances [20].

4.4 All Stabilizing Controllers

The robust stabilizing controller Σ_K, which satisfies equation (4.6) for the H_∞ (sub)optimal control system design, is not unique. By definition, this controller needs to achieve robust stability not only for the nominal plant $\hat{G}(\theta, T)$, but also for all plants in a suitably large neighborhood of the nominal plant $\hat{G}(\theta + \theta_\nabla, T)$. In general, there are infinite number of controllers that can robustly stabilize or maintain stability of an uncertainty model Σ, described by $\hat{G}(\theta, T)$ and Δ. The controller, which internally stabilizes the plant, is called *admissible*. An admissible controller is called *optimal H_∞ controller* Σ_K if it was obtained such that $\|T_K\|_\infty$ is minimized. The closed-form solution to the optimal H_∞ control problem is available only for some special cases, and a simpler problem results if one relaxes the minimization condition and settles for a suboptimal solution [20]. An admissible controller is called *suboptimal H_∞ controller* Σ_K if for a given $\gamma > 0$ there are any controllers such that $\|T_K\|_\infty < \gamma$. In H_∞ control terminology, the set of suboptimal H_∞ controllers, which can robustly stabilize an uncertainty model, is called the *all stabilizing controller* [79-80]. This set of all admissible controllers, which satisfies $\|T_K\|_\infty < \gamma$, equals the set of all transfer-matrices $y_n \to u_n$ that can be represented using LFT with a free parameter Q_∞ as $\Sigma_K(Q_\infty)$, shown in Figure 4-4. The selection of the parameter matrix Q_∞ only needs to satisfy the norm $\|Q_\infty\|_\infty < \gamma$.

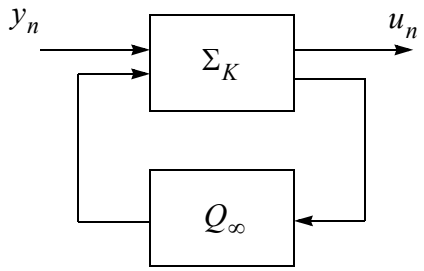


Figure 4-4 Controller Q parametrization

With $Q_\infty = 0$ (at the "center" of the set $\|Q_\infty\|_\infty < \gamma$) the so called *central controller* is selected [80]. This central controller has the alternative interpretation as the controller which minimizes the *entropy integral* [78] defined as [88]

$$I_d(T_K, \gamma) = -\frac{\gamma^2}{2\pi} \cdot \int_{-\pi}^{\pi} \ln\{\det[I - \gamma^{-2} \cdot \tilde{T_K}(e^{j\omega}) \cdot T_K(e^{j\omega})]\} \cdot d\omega \tag{4.8}$$

where $\tilde{T_K}(e^{j\omega}) = T_K{}^*(e^{-j\omega})$. Considering that the central controller minimizes the entropy integral [7] of the closed-loop control system transfer function T_K, it reveals some new properties for the H_∞ (sub)optimal central controller.

First, it can be observed that the entropy integral is finite if, and only if $\gamma^2 \cdot I > \tilde{T_K}(e^{j\omega}) \cdot T_K(e^{j\omega})$, which implies $\gamma > \|T_K\|_\infty$. Thus, the integral is finite if and only if γ is an achievable bound in the H_∞ (sub)optimal control problem [20].

Second, it is shown in [83] that

$$\lim_{\gamma \to \infty} I_d(T_K, \gamma) = \|T_K\|_2^2. \tag{4.9}$$

Therefore, the entropy $I_d(T_K, \gamma)$ is in fact a performance index, measuring the trade-off between the H_∞ optimality $\gamma \to \|T_K\|_\infty$ and the H_2 optimality $\gamma \to \infty$ [83]. It should be remembered at this point that if the controller is designed according to the principles of the H_∞ (sub)optimal control, it is designed to minimize the effects of the peak value of the disturbances affecting the plant, consequently yields to a higher level of robustness. On the other hand, the classical H_2 or Linear Quadratic Gaussian Regulator (LQGR) is designed to minimize the effects of the average value from the disturbances affecting the plant. The H_2 controller is not concerned with the peaks of the disturbances as long as they are acting only for a short time. Consequently it yields to a lower level of robustness. Therefore, it is hoped that by minimizing the entropy function, one can obtain a good trade-off between the H_∞ norm for robustness and the H_2 norm for performance [89].

The properties learned from the entropy function analysis motivate the selection of the central controller for implementation. In recent years, numerous time-domain approaches have emerged to implement the H_∞ control design theory. Among the various approaches to the worst-case design problems, the one that uses the framework of dynamic (differential) game theory seems to be the most natural [19]. In the following chapter, the basics of the game theory relevant to the robust control are discussed.

5 Fundamentals of Game Theory

The game theory was established by the pioneering work of von Neumann[1] in his book *Theory of Games and Economics Behavior,* published in 1944. The game theory is the mathematical formulation of the conflict situation where one or more individuals, the players, have to make decisions and each possible decision leads to a different outcome of the results, which are valued differently by the players. In the following a general introduction is provided on the discrete-time dynamic game theory and its special class, the zero-sum dynamic game. More detailed description can be found in [19] [85].

5.1 Discrete-Time Dynamic Game

If the players can only make moves every T_s time period, and if the game's length is defined as $T_d = T_s \cdot N$, then the game is called discrete-time game of specified fixed duration or finite-horizon control. Using the system theory terminology, the basic elements of a game can be specified by the following definition.

Definition: A K-person discrete-time deterministic dynamic game of specified fixed duration $T_d = N \cdot T_s$ involves [85]:

1. John von Neumann, 1903-1957, Hungarian born mathematician

1. An index set $\mathrm{K} = \{1, \ldots, k, \ldots, K\}$ called the *players' set*, used to distinguish between the actors in the game.
2. An index set $\mathrm{N} = \{0, \ldots, n, \ldots, N\}$ denotes the *stages of the game*, where N is the maximum possible number of moves a player is allowed to make in the game, used as a time measurement.
3. An infinite set X with some topological structure, called the *state set (space) of the game*, to which the state of the game (x_n) belongs for all $n \in \mathrm{N} \cup \{N+1\}$, during the dynamical modelling of the games.
4. An infinite set U_n^k with some topological structures, defined for each $n \in \mathrm{N}$ and $k \in \mathrm{K}$, which is called the *action (control) set* of player P^k at stage n. Its elements are the permissible actions u_n^k of player P^k at stage n. In other words, U_n^k defines the limits of the control action.
5. A function $f_n : X \times U_n^1 \times \ldots \times U_n^K \to X$, defined for each $n \in \mathrm{N}$, so that

$$x_{n+1} = f_n(x_n, u_n^1, \ldots, u_n^K), \quad n \in \mathrm{N} \tag{5.1}$$

for some $x_1 \in X$, which is called the initial state of the game. This difference equation is called the *state equation of the dynamic game*, describing the evolution of the underlying decision process.

6. A set Y_n^k with some topological structure, defined for each $n \in \mathrm{N}$ and $k \in \mathrm{K}$, and called the *observation set* of k^{th} player P^k at stage $n \in \mathrm{N}$, to which the observation y_n^k of player P^k belongs at stage $n \in \mathrm{N}$.

7. A function $h_n : X \to Y_n^k$, defined for each $n \in \mathrm{N}$ and $k \in \mathrm{K}$, so that

$$y_n^k = h_n^k(x_n), \quad n \in \mathrm{N}, \quad k \in \mathrm{K} \tag{5.2}$$

which is the *state-measurement (-observation) equation* of player P^k concerning the value of x_n.

8. A finite set η_n^k defined for each $n \in \mathrm{N}$ and $k \in \mathrm{K}$ as a subset of $\{\{y_1^1, \ldots, y_n^1\}, \ldots, \{y_1^K, \ldots, y_n^K\}, \{u_1^1, \ldots, u_{n-1}^1\}, \ldots, \{u_1^K, \ldots, u_{n-1}^K\}\}$, which determines the information gained and recalled by the player P^k at the stage n of the game. Specification of η_n^k for all $n \in \mathrm{N}$ characterizes the information structure (pattern) of player P^k, and the collection (over $k \in \mathrm{K}$) of these information structures is the *information structure of the game*.

9. A set H_n^k defined for each $n \in \mathrm{N}$ and $k \in \mathrm{K}$ as an appropriate subset of $\{\{Y_1^1, \ldots, Y_n^1\}, \ldots, \{Y_1^K, \ldots, Y_n^K\}, \{U_1^1, \ldots, U_{n-1}^1\}, \ldots, \{U_1^K, \ldots, U_{n-1}^K\}\}$, compatible with η_n^k. H_n^k is called the *information space of the player* P^k at the stage n, induced by his information η_n^k.

10. A prespecified class Γ_n^k of mappings $\gamma_n^k : H_n^k \to U_n^k$ which are the *permissible strategies of player* P^k at stage n. The aggregate mapping $\gamma^k = \{\gamma_1^k, \gamma_2^k, \ldots, \gamma_N^k\}$ is a strategy for the player P^k in the game, and the class Γ^k of all such mappings γ^k so that $\gamma_n^k \in \Gamma_n^k$, $n \in \mathrm{N}$, is the strategy set (space) of the player P^k.

11. A functional

$$L^k : (X \times U_1^1 \times \ldots \times U_1^K) \times (X \times U_2^1 \times \ldots \times U_2^K) \times \ldots \times (X \times U_K^1 \times \ldots \times U_K^K) \to \Re$$

defined for each $k \in \mathrm{K}$, and called the *cost functional of player* P^k in the game of fixed duration.

As a conclusion, for a K player discrete-time dynamic game, for the k^{th} player P^k with strategy set $\{\Gamma^k;\ k \in \mathrm{K}\}$ and with decision (action) vector $\{u^k \in \mathrm{U};\ k \in \mathrm{K}\}$, the player's actions are completely determined by the relations

$$u^k = \gamma^k(\eta^k) \qquad k \in \mathrm{K} \tag{5.3}$$

where the set $\{\gamma^k \in \Gamma^k;\ k \in \mathrm{K}\}$ describes the strategies for the players $\{P^k;\ k \in \mathrm{K}\}$, and η^k is the information available to the player P^k. The optimal strategy set, marked by star, $\{\gamma^{k*} \in \Gamma^k;\ k \in \mathrm{K}\}$, called the *Nash equilibria*, can be calculated using one of the standard techniques of optimal control, like the *minimum principle for open-loop systems* or *dynamic programming for feedback systems*. In game theory, the Nash equilibrium is the synonym for the optimal solution. When the Nash equilibrium is achieved, one player cannot improve his outcome of the game by altering his decision unilaterally. The Nash equilibrium solution has the nice property to guarantee for each player a given optimal value for the objective functional [90].

5.2 Zero-Sum Dynamic Games

A particularly important component in game theory is the conflict situation between two players, where one player's gain incurs a loss to the other player. Due to the conflicting interests of the two players, this game does not allow any cooperation between them. In mathematical terms, the two-player game can be evaluated through its cost functions J^i. If u^i represents the decision variable for the player i with cost function J^i, the sum of the cost functions of the two players [85]

$$J^1(u^1, u^2) + J^2(u^1, u^2) \equiv 0 \tag{5.4}$$

will always be zero (or a constant value), hence the name *two-player zero-sum game.*

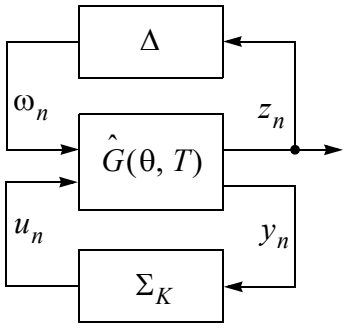


Figure 5-1 Two-player control system model

The control system setup, presented in Figure 5-1, can be regarded as a two-player zero-sum game. In this game, nature Δ is the maximizing player or the opponent. This player has access to the disturbance signal ω_n, which is modulated in a way to maximize

the regulated signal z_n. On the other hand, the controller Σ_K is the minimizing player that is modulating the control signal u_n with a goal to minimize the regulated signal z_n.

Let $J(u, \omega, x)$ be the cost function for this two-player zero-sum game, which is minimized by u, from the possible set of values U, and maximized by ω, from the possible set W, and x is the state-variable of the dynamic system. The game will have an *upper value* $\bar{J}$ if the cost function is first maximized by selecting the optimal value ω^* and after it is minimized by u [85]

$$\bar{J} = J(u, \omega^*, x) = \inf_{u \in U} \sup_{\omega \in W} J(u, \omega, x). \tag{5.5}$$

The game will have a *lower value* $\underline{J}$ if the cost function is first minimized by selecting the optimal value u^* and after it is maximized by ω

$$\underline{J} = J(u^*, \omega, x) = \sup_{\omega \in W} \inf_{u \in U} J(u, \omega, x). \tag{5.6}$$

The optimal solution for the game J^* will have a value between the upper value $\bar{J}$ and lower value $\underline{J}$ of the game [85]

$$J(u^*, \omega, x) \le J^* \equiv J(u^*, \omega^*, x) \le J(u, \omega^*, x) \qquad \forall \omega \in W, \forall u \in U \tag{5.7}$$

where the inequality is called the *Nash inequality.* Furthermore, if an optimal solution exists for the game J^* such that

$$J(u^*, \omega, x) = J(u^*, \omega^*, x) = J(u, \omega^*, x) \tag{5.8}$$

then the Nash equilibria solution pair (u^*, ω^*) is called a *saddle-point* solution or *nonco-operative equilibrium*, and J^* is called the game's *value function.*

The saddle-point solution has the nice property to guarantee to each player a given optimal value for the objective function. For example if (u^*, ω^*) is a saddle-point solution, the choice of $u = u^*$ ensures to player 1 that the value of the objective function is at most equal to $J(u^*, \omega^*)$. The same holds for the player 2 if it chooses $\omega = \omega^*$. Optimality arises from the fact that it is not possible for a player to improve this value if the other player adopts his saddle-point strategy. For this reason $J(u^*, \omega^*, x)$ is called the *value of the game* [90].

5.2.1 Optimal Solution for Discrete-Time Dynamic Game

The discrete-time dynamic process, characterized by its system's state process $\{x_n\}$, can be modelled by its state and output equations, which in general can be nonlinear,

$$x_{n+1} = f_n(x_n, u_n, \omega_n) \qquad n \in [1, N] \tag{5.9}$$

$$y_n = h_n(x_n, \omega_n) \qquad n \in [2, N]. \tag{5.10}$$

If $J(u, \omega)$ denotes the *system's performance criterion*, or *objective function*, for the time interval $N \cdot T_s$, the two-player zero-sum game is defined by the *finite-horizon N-stage cost,* [19]

$$J(u, \omega, x) = p(x_0) + q(x_N) + \sum_{n=1}^{N-1} g_n(x_{n+1}, u_n, \omega_n, x_n) \tag{5.11}$$

where $p(x_0)$ is the initial cost, $q(x_N)$ is the final cost and $g_n(x_{n+1}, u_n, \omega_n, x_n)$ is the cost at the time $t = n \cdot T_s$. The game is to be minimized by the control policy $\mu = \{u_n\}$ and maximized by the opponent's policy $\nu = \{x_0, \omega_n\}$, both from their respective sets of possible values, $u_n \in U \subset R^r$ and $\omega_n \in W \subset R^l$. The controller's problem is to find a *minimax* strategy $\mu^* = \{u_n^*\}$, more precisely, the strategy that guarantees the best performance, measured by the objective function, in the worst possible situation. The proper strategy minimizes the maximum expected objective function, where the maximum is obtained by the opponent's $\nu^* = \{x_0, \omega_n^*\}$ best policy over all possible strategies ν. To determine its policy $\mu = \{u_n\}$ the controller can access the current as well as the entire past values of the state [19]

$$u_n = \mu_n(x_{[1, \ldots n]}) \tag{5.12}$$

where $\mu_1, \mu_2, \ldots \mu_n$ are the control policies in time.

The control policy, if it exists, can be determined using the *Dynamic Programming* method. The dynamic programming theory is based on Bellman's *principle of optimality*, where the solution is obtained in retrograde time. The counterpart of this in two-player zero-sum dynamic games is a recursive equation that involves (again in retrograde time) the saddle-point solution of the static games, known as the *Isaacs*[1] *equation*. Such an

equation provides a sufficient condition for the existence of a saddle-point for a dynamic game.

To utilize the Isaacs equation, the *value function* of the game is defined as the saddle-point solution of the cost function (5.11), if it exists,[19]

$$\begin{aligned} V_n(x_n) &= \min_{u \in U} \max_{\omega \in W} \left\{ q(x_N) + \sum_{k=n}^{N} g_k(x_{k+1}, u_k, \omega_k, x_k) \right\} \\ &= \max_{\omega \in W} \min_{u \in U} \left\{ q(x_N) + \sum_{k=n}^{N} g_k(x_{k+1}, u_k, \omega_k, x_k) \right\} \end{aligned} . \tag{5.13}$$

Equation (5.13), written as a backward recursion, led to a familiar solution in dynamic programming, called the Isaacs equation

$$\begin{aligned} V_n(x_n) &= \min_{u \in U} \max_{\omega \in W} [g_n(x_{n+1}, u_n, \omega_n, x_n) + V_{n+1}(x_{n+1})] \\ &= \max_{\omega \in W} \min_{u \in U} [g_n(x_{n+1}, u_n, \omega_n, x_n) + V_{n+1}(x_{n+1})] \\ &= g_n(x_{n+1}, u_n^*, \omega_n^*, x_n) + V_{n+1}(x_{n+1}^*) \end{aligned} . \tag{5.14}$$

$$V_{N+1}(x_{N+1}) \equiv 0$$

(5.14) is the Isaacs equation, being the discrete-time version of a similar equation obtained by Isaacs as a solutions for a continuos-time two-player zero-sum dynamic game, which is known as the Hamilton-Jacobi-Isaacs equation [19][85].

---

1. Rufus Philiph Isaac, 1914 - , USA mathematician

By utilizing this backward recursion, the control policies for the minimizing player $u_n^* = \mu^*(x_n)$ and for the maximizing player $\omega_n^* = \nu^*(x_n)$ are calculated from the current value of the state x_n. This is known as the feedback policy. If the equivalence in (5.14) exists, it is a saddle-point of the game and the corresponding performance index, the value of the game, is given by $V_1(x_1)$.

The control policies $u_n^* = \mu^*(x_n)$ assume that the minimizing player has full knowledge of the state-variables x_n, called the *full-information control*. This situation rarely occurs. In most cases the minimizing player can access only the measurement sequence $y_1, \ldots, y_{n-1}$ up to the current time $t = nT_s$ from the plant output $y_n = h_n(x_n, \omega_n)$ as defined in (5.10). This type of control is the *measurement-feedback control*.

A similar problem is solved by the Linear Quadratic Gaussian Regulator (LQGR) design using the *separation principle*. In LQGR the control is separated into an observer and a linear feedback structure for the control $u_n = K \cdot \hat{x}_n$. The observer estimates the state-variables $\hat{x}_n$ using the Kalman Filter. The main difference between the LQGR and the minimax design is that in the latter no separation principle applies. In minimax design, the observer and the controller are linked [82]. Nevertheless, the solution to the $u_n^* = \mu^*(x_n)$ controller synthesis problem may be obtained by solving the full-information control problem and finding a minimax estimator for the full-information control [7]. The two processes are not separated but rather coupled in a way that the state estimation filter design depends on the solution for control, but not vice versa [20]. The theory that

upholds the state estimation filter design for the minimax controller is called the *certainty equivalence principle*, which is the topic of the following section.

5.3 Certainty Equivalence Principle

As shown above, the players using the current value of the state x_n are evaluating the game with the value function $V_n(x_n)$, and calculating their optimizing feedback controls $u_n^* = \mu^*(x_n)$, $\omega_n^* = \nu^*(x_n)$. The difficulty with this full state feedback control strategy is that in most cases neither player has access to the exact value of the state vector [91]. Due to the lack of access to the x_n state-variables, the players can not evaluate their optimal saddle-point strategies, which brings uncertainty into the game. To resolve this problem, a risk-sensitive state estimator that may be fed into the full state feedback control is introduced in the form of the certainty equivalence principle [19][91][92].

Assume that an optimal control policy μ^* is required to choose a control $u_n \in U \subset R^r$ to minimize the cost function $J(u, \omega, x)$. In the case of perfect information where the system states $x_n \in X \subset R^m$ are known, the optimal control policy $u_n^* = \mu^*(x_n)$ is chosen such that $J(u^*(x), \omega, x) \leq J(u, \omega, x)$ for all $u_n, n = 1, 2 \ldots N$. However, if x_n is only known through the observations of the plant output $y(x_n)$, and the solution of the optimal control policy $u_n^* = \mu^*(y)$ is not known, it is not totally without merit to use the control law [19]

$$u_n^* = \hat{\mu}(y) = \mu^*(\hat{x}_n) \tag{5.15}$$

where

$$\hat{x}_n \Leftarrow T(u_{n-1}, y_{n-1}, \hat{x}_n) \tag{5.16}$$

under the condition that there is a function $T(u_{n-1}, y_{n-1})$, which is an estimator for the system states x_n, constructed on the observed sequences $\{u_1, \ldots, u_{n-1}\}$ and $\{y_1, \ldots, y_{n-1}\}$. This idea, called *certainty equivalence principle*, states that an optimal controller can be obtained by inserting an estimate (if one exists) into the corresponding optimal state feedback law [92].

The measurement-feedback control problem can be solved via a two-step procedure. In the first step, a full-information control problem is solved assuming that the states of the system are known. The second step entails the estimation of the unknown unmeasurable system states, using the observed system output measurements.

In the selection of the system state estimation function $T(u_{n-1}, y_{n-1}, \hat{x}_n)$, using the game theoretic approach, the goal is to anticipate the "worst possible disturbances" generated by nature, the maximizing player. Accordingly, the aim is to estimate the "worst possible current state" $\hat{x}_n$ compatible with the available information. If this state estimate is used by the minimizing player, the worst possible action of the maximizing player will be minimized.

Applying this principle to deterministic games under partial information, the certainty equivalence implies that one should use the optimal control corresponding to the state estimate $\hat{x}_n$, obtained by maximizing the performance index [91]

$$T(u_{n-1}, y_{n-1}, \hat{x}_n) = \max_{x_n \in X} [L(\hat{x}_n) + V(\hat{x}_n)] \tag{5.17}$$

where $L(\hat{x}_n)$ is the information state and $V(\hat{x}_n)$ is the value function, defined by (5.14). The information state $L(\hat{x}_n)$ is essentially the worst-case cost-so-far because it is evaluated in the past from the initial time to the present time.

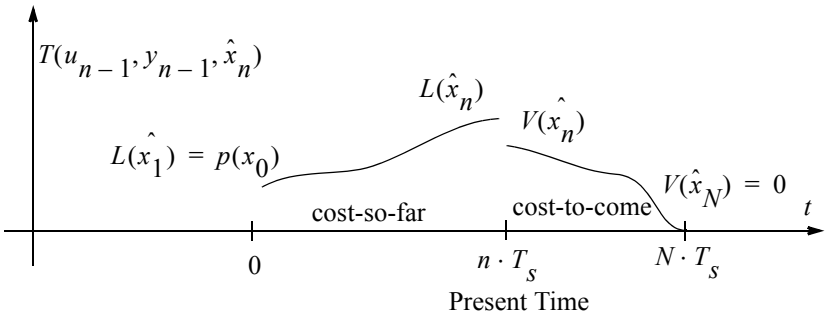


Figure 5-2 Performance index for the game

The value function $V(\hat{x}_n)$ is the minimax cost-to-come, since it is evaluated in the future, backward, from the final time to the present. So, heuristically, this is roughly equivalent to taking the worst-case possibility for total cost from initial time to final time [91]. Figure 5-2 serves as an illustration of the components of the performance index $T(u_{n-1}, y_{n-1}, \hat{x}_n)$.

To anticipate the worst possible influence of the maximizing player, the general form of information state $L(\hat{x}_n)$ as a worst past cost, can be defined as a function of $\{u_1, \ldots, u_{n-1}\}$ and $\{y_1, \ldots, y_{n-1}\}$ [19]

$$L(\hat{x}_n) = \sup_{(x_0, \omega_{k-1}) \in \Omega_{n-1}} \left[p(x_0) + \sum_{k=1}^{n-1} g_k(\hat{x}_{k+1}, y_k, u_k, \omega_k) \right] \tag{5.18}$$

where, $p(x_0)$ is the cost on the initial state and $g_k(\hat{x}_{k+1}, y_k, u_k, \omega_k)$ is the cost on the state at time $t = k \cdot T_s$. Since only partial information is available from the state, the initial state x_0 is also unknown. This problem can be handled by considering x_0 as part of the disturbance, controlled by the maximizing player, included within the initial cost term $p(x_0)$.

Furthermore, Ω_{n-1} represents the set of all possible (x_0, ω) disturbances consistent with the observations from $\{u_1, \ldots, u_{n-1}\}$ and $\{y_1, \ldots, y_{n-1}\}$ such that the equations (5.9) and (5.10) are satisfied. Thus, x_0 and the past values of ω_{n-1} should agree with the past control and measurements but future ω_n could be the worst possible, if it belongs to the action (control) set of the maximizing player and it is chosen according to $\nu^*(\hat{x}_n)$, leading to a least cost-to-go $V(\hat{x}_n)$ [19].

Using a simple recursion, the information state $L(\hat{x}_n)$ can be written in a forward dynamic programming equation form

$$L(\hat{x}_n) = \sup_{\omega_{n-1} \in W} [g_n(\hat{x}_n, y_{n-1}, u_{n-1}, \omega_{n-1}) + L(\hat{x}_{n-1})] \tag{5.19}$$

where the recursion is initialized by $L(\hat{x}_1) = p(x_0)$. If the recursion satisfies the saddle-point solution requirements, which is formulated below, it can be regarded as the general definition for the state estimator.

The state estimator, obtained as an upper limit of the recursion (5.19), is only the necessary condition for the existence of one. Defined by the certainty equivalence principle in (5.17), the sufficient condition for the worst-case state estimator is the existence of the saddle-point solution for the equation (5.17). This sufficient condition is formulated next.

If the value function $V(\hat{x}_n)$ in the performance index (5.17) is replaced by the one based on the Isaacs equation, as defined in (5.14), equation (5.17) can be rewritten as [19]

$$T(u_{n-1}, y_{n-1}, \hat{x}_n) = \max_{x_n \in X} \min_{u_n \in U} \max_{\omega_n \in W} [g_n(x_n, u_{n-1}, \omega_{n-1}, x_{n-1}) + V_{n+1}(x_{n+1}) + L(\hat{x}_{n-1})] \quad . \tag{5.20}$$

If there is a solution to the two-players game, $T(u_{n-1}, y_{n-1}, \hat{x}_n)$ is required to be a saddle-point solution

$$T(u_{n-1}, y_{n-1}, \hat{x}_n) = \max_{x_n \in X} \min_{u_n \in U} \max_{\omega_n \in W} [g_n(x_n, u_{n-1}, \omega_{n-1}, x_{n-1}) + V_{n+1}(x_{n+1}) + L(\hat{x}_{n-1})]$$

$$= \min_{u_n \in U} \max_{x_n \in X} \max_{\omega_n \in W} [g_n(x_n, u_{n-1}, \omega_{n-1}, x_{n-1}) + V_{n+1}(x_{n+1}) + L(\hat{x}_{n-1})] \quad . \tag{5.21}$$

If the saddle-point solution exists then it yields to a uniquely defined $u_n^* = \mu^*(y)$, which is a minimax controller [19].

The intuition behind introducing this controller, using the certainty equivalence principle, can be interpreted in the following way. At time $t = n \cdot T_s$, knowing the observations $y_1, \ldots, y_{n-1}$ and the control $u_1, \ldots, u_{n-1}$ up to time $t = (n-1) \cdot T_s$, one should anticipate the "worst disturbance" ω_n compatible with the available information and in accordance with the performance index (5.17). Then estimate the corresponding "worst possible current state" $\hat{x}_n$, by (5.19), and use it as if it were indeed the actual current state for the control. This is indeed a worst-case certainty-equivalence principle, but it is not a separation principle since the worst-case state estimate depends on the performance index (5.17) to be minimized [19].

5.4 Finite-Horizon Discrete-Time Linear System

The previously developed general discrete-time control theory can be adapted to the situation where the system is considered to be finite-horizon discrete-time linear, and the performance index for the control function is quadratic. In this setup, it is required that the signals belong to the $l_{2,[0,N]}$ space, see Appendix D. The plant model Σ, a discrete-time

linear state space equation, is defined by the difference state equation and by the two output equations

$$\Sigma : \quad \begin{aligned} x_{n+1} &= A \cdot x_n + B \cdot u_n + D \cdot \omega_n \\ y_n &= C \cdot x_n + E \cdot \omega_n \\ z_n &= H \cdot x_n + G \cdot u_n \end{aligned} \tag{5.22}$$

where $x_n \in \Re^m$ represents the state variable, $\omega_n \in \Re^l$ is the disturbance (control input of the maximizing player), $u_n \in \Re^r$ is the control input (control input of the minimizing player), $y_n \in \Re^q$ is the measurement output, $z_n \in \Re^q$ is the controlled output, and A, B, D, C, E, H, G are (possible time-variables) matrices with appropriate dimensions.

The design of the worst-case control system Σ_K, using the H_∞ (sub)optimal control system design process, is based on the requirements stipulated by the inequality (4.6). Using the relations of (4.7) and by relaxing the inequality of (4.6), the main requirement for the H_∞ (sub)optimal control is formulated as

$$\|z\|^2_{l_2[0,\,N-1]} - \gamma^2 \cdot \|\omega\|^2_{l_2[0,\,N-1]} \le 0 \qquad \text{for all } \omega \in l_2 \tag{5.23}$$

for some prespecified positive real number γ. This can be regarded as the main idea for the worst-case controller cost function definition. In the performance index definition, the norm operator can be substituted with the sum of squares. Further, equation (5.23) can be extended with the cost on the initial and final states. It is assumed that the initial state x_0 is not known for the minimizing player and that it is part of the disturbance, controlled by the maximizing player. The cost on the final state x_N expresses the desire for the system

behavior at the end of the control horizon. Taking all these premises into consideration, a control system for the plant can be designed, if a saddle-point solution exists for the associated soft-constrained linear quadratic game with the objective function [19][20]

$$J_\gamma(u, \omega) = x_N^T \cdot X^C \cdot x_N + \sum_{n=1}^{N-1} \left\{ z_n^T \cdot z_n - \gamma^2 \cdot \omega_n^T \cdot \omega_n \right\} - \gamma^2 \cdot x_0^T \cdot X^F \cdot x_0 \qquad (5.24)$$

for some $\gamma \geq 0$ where $X^C \geq 0$, and $X^F \geq 0$. In this game, the minimizing player controls u_n, calculating its strategy based on the observations of the measurement output. It is expected that the resulting admissible controller is strictly causal output feedback of the form $u_n = \mu(u_{n-1}, y_{n-1})$. The maximizing player controls the disturbances ω_n either as a state dependent or in an open-loop fashion [93].

The design of the measurement feedback H_∞ (sub)optimal controller consists of three steps. In the first step a controller is designed assuming a full state feedback. The solution for the game is obtained if the game has as a saddle-point solution $J_\gamma(u^*, \omega^*)$ (5.5)-(5.8)

$$J_\gamma(u^*, \omega^*) = \min_{u \in U} \max_{\omega \in W} J_\gamma(u, \omega) = \max_{\omega \in W} \min_{u \in U} J_\gamma(u, \omega). \qquad (5.25)$$

Then, the value function is defined as

$$V_n(x_n) = \min_{u \in U} \max_{\omega \in W} \left\{ x_N^T \cdot X^C \cdot x_N + \sum_{k=n}^{N-1} \left\{ z_n^T \cdot z_n - \gamma^2 \cdot \omega_n^T \cdot \omega_n \right\} \right\} \qquad (5.26)$$

and the associated Isaacs equation is

$$V_n(x_n) = \min_{u \in U} \max_{\omega \in W} \left\{ \left\{ z_n^T \cdot z_n - \gamma^2 \cdot \omega_n^T \cdot \omega_n \right\} + V_{n+1}(x_{n+1}) \right\} \tag{5.27}$$

with the initializing condition $V_0(x_0) = 0$. For the linear-quadratic game the suitable value function $V_n(x_n)$, which for the given saddle-point policy uniquely solves the Isaacs equation (5.14), is defined as [19]

$$V_n(x_n) = x_n^T \cdot P_n^C \cdot x_n \tag{5.28}$$

where the matrix P_n^C needs to be determined.

Using the equations (5.27) and (5.28)

$$x_n^T \cdot P_n^C \cdot x_n = \min_{u \in U} \max_{\omega \in W} \left\{ \left\{ z_n^T \cdot z_n - \gamma^2 \cdot \omega_n^T \cdot \omega_n \right\} + x_{n+1}^T \cdot P_{n+1}^C \cdot x_{n+1} \right\} \tag{5.29}$$

and substituting x_{n+1} and z_n from (5.22), leads to the following equation

$$\begin{aligned} x_n^T \cdot P_n^C \cdot x_n = \min_{u \in U} \max_{\omega \in W} \Big\{ & (H \cdot x_n + G \cdot u_n)^T \cdot (H \cdot x_n + G \cdot u_n) + \\ & -\gamma^2 \cdot \omega_k^T \cdot \omega_k + \\ & +(A \cdot x_n + B \cdot u_n + D \cdot \omega_n)^T \cdot P_{n+1}^C \cdot (A \cdot x_n + B \cdot u_n + D \cdot \omega_n) \Big\} \end{aligned} \tag{5.30}$$

The solution of the above equation defines the optimal feedback control policy, where P_n^C, if it exists, is attained as a solution of the well known backward Riccati recursion from dynamic programming.

The second step entails the design of the worst-case state estimator, using the certainty equivalence principle. The state estimate $\hat{x}_n$, which is obtained by maximizing the performance index (5.17), repeated hereafter for convenience

$$T(u_{n-1}, y_{n-1}, \hat{x}_n) = \max_{x_n \in X} [L(\hat{x}_n) + V(\hat{x}_n)] \tag{5.31}$$

where $V_n(\hat{x}_n)$ is defined by (5.27) and the information state $L_n(\hat{x}_n)$ is calculated by

$$L_n(\hat{x}_n) = \sup_{\omega_k \in \Omega_{n-1}} \left\{ \sum_{k=1}^{n-1} \left\{ z_n^T \cdot z_n - \gamma^2 \cdot \omega_n^T \cdot \omega_n \right\} - \gamma^2 \cdot x_0^T \cdot X^F \cdot x_0 \right\} \tag{5.32}$$

where Ω_{n-1} is the set of all possible $\{x_0, \omega_{k,[1,n-1]}\}$ disturbances, consistent with the observations of $\{u_{k,[1,n-1]}\}$ and $\{y_{k,[1,n-1]}\}$. The information state $L_n(\hat{x}_n)$ can be written in a forward dynamic programming equation form

$$L(\hat{x}_n) = \sup_{\omega_{n-1} \in \Omega_{n-1}} \left[\left\{ z_{n-1}^T \cdot z_{n-1} - \gamma^2 \cdot \omega_{n-1}^T \cdot \omega_{n-1} \right\} + L(\hat{x}_{n-1}) \right] \tag{5.33}$$

where the recursion is initialized by $L(\hat{x}_0) = -\gamma^2 \cdot x_0^T \cdot X^F \cdot x_0$.

For the linear-quadratic game the suitable information state $L_n(\hat{x}_n)$, which for the given saddle-point policy uniquely solves the Isaacs equation (5.33), is defined as [19]

$$L(\hat{x}_n) = -\gamma \cdot [x_n - \hat{x}_n]^T \cdot P_n^F \cdot [x_n - \hat{x}_n] + \phi_n \tag{5.34}$$

where the vector $\hat{x}_n$, the matrix P_n^F and the scalar ϕ_n are coefficients to be determined.

Substituting (5.34) in (5.33)

$$-\gamma \cdot [x_n - \hat{x}_n]^T \cdot P_n^F \cdot [x_n - \hat{x}_n] + \phi_n \;\; =$$

$$\sup_{\omega_{n-1} \in \Omega_{n-1}} \left\{ \left\{ z_{n-1}^T \cdot z_{n-1} - \gamma^2 \cdot \omega_{n-1}^T \cdot \omega_{n-1} \right\} + \right. \tag{5.35}$$

$$\left. -\gamma \cdot [x_{n-1} - \hat{x}_{n-1}]^T \cdot P_{n-1}^F \cdot [x_{n-1} - \hat{x}_{n-1}] + \phi_{n-1} \right\}$$

takes the form of the information state equation, the solution of which defines the worst-case state estimation policy. In this equation P_n^F, if it exists, is attained as a solution of the well known forward Riccati recursion from dynamic programming.

The final step in the design of the measurement feedback H_∞ (sub)optimal controller is to test the coupling condition between the control and the state estimator. The condition outlined by the solution of equation (5.31) leads to the so called coupling condition

$$(\gamma^2 \cdot I - P_n^F \cdot P_n^C) > 0 \tag{5.36}$$

which expresses the coupling requirements towards the solution of the control in step one and towards the solution of the state estimator in step two.

In summary, the resulting game-theoretic saddle-point solution for robust control can be interpreted as follows: having worked out the worst possible case noise/perturbation that nature could impose given the objective, it is still practicable to design an optimal controller for achieving stability and disturbance attenuation. The H_∞ full-information controller accomplishes this task. Once completed, an optimal estimator of actual noise

and perturbation can also be designed quite independently. This is the H_∞ estimator for the accompanying full-information controller. However, due to a coupling between the control and estimation problems, the filter cannot be designed independently of the control objective and depends on the relevant control law. Of necessity, the relevant Riccati equations must also have feasible solutions [94].

The main ideas for a worst-case control system design using the two-player zero sum game as a theoretical background to define the H_∞ (sub)optimal control algorithm are described in this chapter. The detailed solution and implementation of a two-player zero-sum linear-quadratic game is presented in the following chapter.

6 Measurement Feedback H_∞ Controller

Robust control problems are often used to solve the minimax type two-player zero-sum dynamic games. For the linear systems with quadratic cost criteria and with the assumption that all the state variables are known, the controller is obtained with the aid of a Riccati equation. In the partially observed case, when only the plant outputs are observable, the certainty equivalence principle certifies the use of the observed states instead of the measured one. The observer calculates the states of the system from the observation of the past plant output. This observer is obtained with the aid of a second Riccati equation.

This chapter describes the practical implementation of the linear discrete-time measurement feedback H_∞ (sub)optimal control algorithm. As a theoretical base for this algorithm, the game theoretical approach, as defined in the previous chapter (Section 5.4), is used [95-100].

6.1 Standard H_∞ Control Problem

The widely accepted, so called standard control system, representation for the H_∞ control system can be obtained using the ideas introduced in Figure 4-2. The standard control system, as presented in Figure 6-1, is obtained by expanding the linear state space plant model with ω_n input and z_n output.

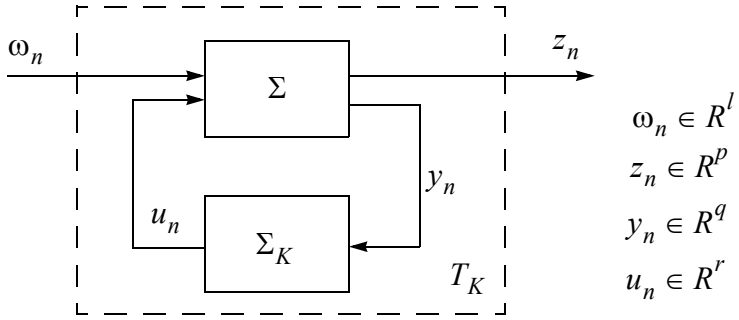


Figure 6-1 Measurement feedback control system T_K

The ω_n input can be considered as exogenous, which includes disturbances and commands. The z_n output, the so called regulated output, is used to express robustness and performance requirements in control system design. The linear discrete-time finite-dimensional system Σ, in state-space realization form can be formulated as [19]

$$\Sigma: \quad \begin{aligned} x_{n+1} &= A \cdot x_n + B \cdot u_n + D \cdot \omega_n \\ y_n &= C \cdot x_n + E \cdot \omega_n \\ z_n &= H \cdot x_n + G \cdot u_n \end{aligned} \tag{6.1}$$

where for all $n \in \aleph$, $x_n \in \Re^m$ is the state, $u_n \in \Re^r$ is the control input, $y_n \in \Re^q$ is the measurement, $\omega_n \in \Re^l$ is the unknown disturbance and $z_n \in \Re^p$ is the output to be controlled. $A \in \Re^{m \times m}$, $B \in \Re^{m \times r}$, $C \in \Re^{q \times m}$, $D \in \Re^{m \times l}$, $E \in \Re^{q \times l}$, $H \in \Re^{p \times m}$ and $G \in \Re^{p \times r}$ are the matrices of appropriate dimensions, which in general are time varying. The matrices A, B and C are defined during the system modelling. On the other hand, the matrices D, E, H and G are selected to express robustness and performance require-

ments for the control system. This selection is mostly done by iteratively solving and implementing a number of H_∞ control problems, until the desired performance is achieved [101]. The goal during the selection of the weights is to minimize the effect of the disturbances ω_n on the controlled output z_n by finding a dynamic feedback law which calculates the appropriate control input u_n. The resulting H_∞ control system design leads to a dynamic feedback law in state-pace form [19]

$$\Sigma_K: \quad \begin{aligned} \hat{x}_{n+1} &= F^x \cdot \hat{x}_n + F^u \cdot u_n + F^y \cdot y_n \\ u_n &= K^x \cdot \hat{x}_n \end{aligned} \tag{6.2}$$

where F^x, F^u, F^y and K^x are matrices with appropriate dimensions, calculated from Σ by the H_∞ design algorithm. This dynamic feedback operator Σ_K minimizes the l_2-induced operator norm of $\omega \to z$, that internally stabilizes the plant.

The state-space formulation of the closed-loop system T_K, with $x_n, \hat{x}_n$ state-variables is

$$T_K: \quad \begin{aligned} \begin{bmatrix} x_{n+1} \\ \hat{x}_{n+1} \end{bmatrix} &= \begin{bmatrix} A & B \cdot K^x \\ F^y \cdot C & F^u \cdot K^x + F^x \end{bmatrix} \cdot \begin{bmatrix} x_n \\ \hat{x}_n \end{bmatrix} + \begin{bmatrix} D \\ F^y \cdot E \end{bmatrix} \cdot \omega_n \\ z_n &= \begin{bmatrix} H & G \cdot K^x \end{bmatrix} \cdot \begin{bmatrix} x_n \\ \hat{x}_n \end{bmatrix} \end{aligned} . \tag{6.3}$$

This closed-loop system T_K is internally stabilized by the controller Σ_K. The closed-loop system is internally stable if, with $\omega = 0$ and for all initial states x_0, the state converges to zero $z_n \to 0$ as $n \to \infty$ [78]. The system (6.3) is internally stable if the $2m \times 2m$

closed-loop system-matrix (6.4) is asymptotically stable. See “Stability of Discrete-Time Systems” on page 312.

$$\bar{A} = \begin{bmatrix} A & B \cdot K^x \\ F^y \cdot C & F^u \cdot K^x + F^x \end{bmatrix}. \tag{6.4}$$

6.2 Finite Horizon Suboptimal H_∞ Controller

For the plant model Σ, defined by equation (6.1), the central controller Σ_K (6.2) can be found if the associated soft-constrained linear-quadratic game with objective function, as introduced in (5.24) has a feedback saddle-point solution for some pre-selected γ [19]

$$J_\gamma(u, \omega) = x_N^T \cdot X^C \cdot x_N + \sum_{n=1}^{N-1} \left\{ \begin{bmatrix} x_n^T & u_n^T \end{bmatrix} \cdot \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \cdot \begin{bmatrix} x_n \\ u_n \end{bmatrix} - \gamma^2 \cdot \omega^T \cdot \omega \right\} - \gamma^2 \cdot x_0^T \cdot X^F \cdot x_0 . \tag{6.5}$$

The concept of γ, as a measure for the control system robustness is defined in Chapter 4, and the selection of its value is clarified in Section 6.2.2. Where γ is a scalar that determines the respective contributions of the original quadratic cost, z_n in (6.1), and disturbance energy to the cost function [20]. Further, $X^C \geq 0$ is the weighting matrix for the final state, see (6.20) and the comments thereafter. The weighting matrix for the initial state is $X^F \geq 0$, see (6.24) and the comments thereafter. Finally, Q, R and S are weight-

ing matrices for the states, for the control and for the combined states and control, respectively, defined by (6.14).

6.2.1 Preliminary Assumptions for the Control

For the existence of the linear discrete-time measurement feedback H_∞ (sub)optimal control algorithm, Σ_K (6.2), it is expected that the following necessary requirements are satisfied.

Controllability condition. The existence of the stabilizing controller requires that the unstable modes have to be reachable by the control signal u_n, or equivalently the pair $\{A, B\}$ in (6.1) needs to be *controllable* over $[n, N), \forall n \in (0, N)$. See "Controllability of Discrete-Time Systems" on page 313.

Observability condition. The controller is required to have access to all the information concerning the unstable modes, the pair $\{A, C\}$ needs to be *detectable* over $[0, n), \forall n \in (0, N]$. See "Observability of Discrete-Time Systems" on page 314.

Injective condition for G. The matrix G is required to have *full column rank* in order that all control signals u_n in (6.1) be penalized, or equivalently matrix G needs to be injective (one to one)

$$\text{if} \quad G \in \Re^{p \times r} \quad \text{and} \quad c_G = \text{rank}(G) \qquad \text{then} \qquad c_G \Leftrightarrow r \quad . \tag{6.6}$$

Surjective condition for E. The matrix E in (6.1) is required to have *full row rank* in order to characterize the measurement noise. Therefore, for each output signal y_n some disturbance shall be defined, and modelled as direct feed-through from the disturbance signal ω_n in (6.1). Noise-free measurements can not exist as they would be infinitely reliable. In other words, the matrix E needs to be surjective (onto)

$$\text{if} \quad E \in \Re^{q \times l} \quad \text{and} \quad r_E = \text{rank}(E) \qquad \text{then} \qquad r_E \Leftrightarrow q \quad . \tag{6.7}$$

The assumptions for the injective and surjective conditions require the dimensions of the controlled output $z_n \in \Re^p$ to be at least that of the control input $u_n \in \Re^r$ or $p \geq r$, while the dimension of the unknown disturbance $\omega_n \in \Re^l$ to be at least that of the measurement $y_n \in \Re^q$, or $l \geq q$. Since no measurement is error free and no control action is costless, these are reasonable assumptions [7].

Surjective condition for [A D]. The matrix $\begin{bmatrix} A & D \end{bmatrix}$ in (6.1) is required to have *full row rank*

$$\text{rank} \begin{bmatrix} A & D \end{bmatrix} = m \, . \tag{6.8}$$

This requirement is equivalent to the statement that for a sequence $\{u_n\}$, the map $(x_0, \{\omega\}_{[0,n]}) \to x_{n+1}$ be surjective (onto) for all n, which is equivalent to the requirement that the system (6.1) is reachable by ω.

Injective condition for $[\mathbf{A^T}\ \mathbf{H^T}]^\mathbf{T}$. The matrix $\begin{bmatrix} A^T & H^T \end{bmatrix}^T$ in (6.1) is required to have *full column rank*

$$\operatorname{rank}\begin{bmatrix} A \\ H \end{bmatrix} = m. \tag{6.9}$$

This requirement is equivalent to the statement that for a sequence $\{u_n\}$ and $\{\omega_n\}$, the map $x_0 \to (\{z\}_{[0,n]}, x_{n+1})$ be injective (one to one) for all n, which is equivalent to the requirement that the system (6.1) is observable by z.

6.2.2 Selection of γ

The theoretical background for interpreting γ as an upper bound $(1/\gamma)$ for the uncertainties of the control system model is detailed in Chapter 4. In this section a practical selection of this parameter γ is presented. From the lower side this γ is bounded by the infimum γ_0, the H_∞ norm of the transfer function $T_K\colon \omega \to z$. Unfortunately, there is no analytical expression which would calculate the value γ_0. On the other hand, the upper side of the γ is soft bounded by the vanishing robustness in the controller's performance.

From the technical point of view, it is essential that a sufficiently large γ be chosen in order to obtain an admissible controller but to achieve the highest performance in the robustness of the control loop, γ must be close to the infimum γ_0

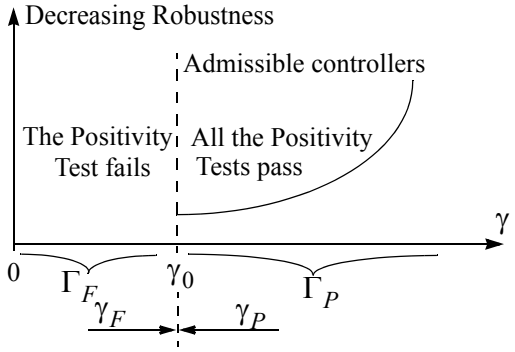


Figure 6-2 Infimum search

The above discussion can be summarized by the definition of $\Gamma_F : \gamma \in [0 : \gamma_0]$ and $\Gamma_P : \gamma \in (\gamma_0 : \infty]$, as presented in Figure 6-2.

The following two statements are equivalent:

- If γ belongs to Γ_P, the H_∞ (sub)optimal control system design leads to an admissible control system, and if γ belongs to Γ_F the controller in not admissible.
- The controller is admissible if the positivity tests (6.23), (6.27) and (6.30) or (6.31), given as an integral part of the H_∞ algorithm are satisfied. If any of the three positivity tests fails the controller is not admissible.

These statements then lead to the conclusion that in order to obtain the minimally achievable H_∞ norm γ_0, it is required to interactively check the positivity tests for different values of γ by a search algorithm.

The proposed algorithm, as presented in Figure 6-3, starts by selecting any positive number for γ and initializing the temporary variables $\gamma_F = 0$ and $\gamma_P = 0$, which are the containers for the γ if it fails or passes the positivity tests, respectively, see Figure 6-2. During the search, if for the selected value of γ, the H_∞ control optimization algorithm passes all three positivity tests then $\gamma \in \Gamma_P$ and $\gamma_P = \gamma$ otherwise $\gamma \in \Gamma_F$ and $\gamma_F = \gamma$. The search algorithm consists of two phases. The first phase lasts until a $\gamma \in \Gamma_P$ is found by the binomial search

$$\gamma = 2 \cdot \gamma . \tag{6.10}$$

The first phase yields to a lower limit γ_F and upper limit γ_P on the infimum γ_0. The second phase is the bisectional search for γ_0 to obtain a close to optimal solution

$$\gamma = (\gamma_P + \gamma_F)/2 . \tag{6.11}$$

This iterative procedure continues until the magnitude of the difference between the smallest γ_P and the largest γ_F is small [102]

$$\text{if } \left(\frac{\gamma_P - \gamma_F}{\gamma_P} < \gamma_{TOL}\right) \quad \text{then} \quad \gamma_0 = \gamma_P \tag{6.12}$$

where the tolerance $\gamma_{TOL} > 0$ is selected according to the required accuracy for the optimal solution of the infimum γ_0.

According to the literature [78] using approximately 10% larger γ than the infimum γ_0 for the H_∞ control system design, leads to a proper solution

$$\gamma = 1.1 \cdot \gamma_0 . \tag{6.13}$$

Otherwise, numerical difficulties will be encountered. The required matrix inverse calculation $(\gamma^2 \cdot I_m - P_n^F \cdot P_n^C)^{-1}$, which appears in almost all formulations for the H_∞ control, is the reason for the numerical difficulties near the infimum γ_0 [101]. The closer the iteration gets to the infimum $\gamma \to \gamma_0 = sup\sqrt{\rho(P_n^F \cdot P_n^C)}$, the more the matrix inverse approaches singularity.

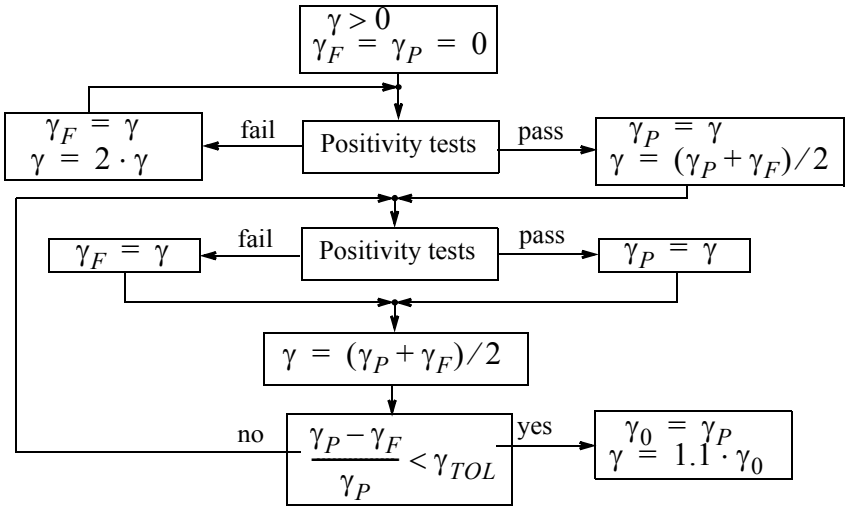


Figure 6-3 Search algorithm for γ

6.2.3 The H_∞ Algorithm

The necessary and sufficient condition for the existence of an admissible controller involves the existence of two positive definite matrices, which are formed from the stabi-

lizing solutions of two Riccati recursions. Moreover, the coupling condition between the solutions of these two Riccati recursions must be satisfied.

The algorithm becomes more readable by the introduction of the notation

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \begin{bmatrix} H^T \\ G^T \end{bmatrix} \cdot \begin{bmatrix} H & G \end{bmatrix} \tag{6.14}$$

where Q represents the weight on the state variable x, R is the weight for the control signal u and S is the weight on the cross terms between the state and the control signal $x \cdot u$. Furthermore,

$$\begin{bmatrix} W & L \\ L^T & V \end{bmatrix} = \begin{bmatrix} D \\ E \end{bmatrix} \cdot \begin{bmatrix} D^T & E^T \end{bmatrix} \tag{6.15}$$

where W represents the measure of system noise intensity and V is the measure of measurement noise intensity [19].

The following notations are introduced for further simplification of the formulas

$$\overline{A} = A - B \cdot R^{-1} \cdot S^T \tag{6.16}$$

$$\overline{Q} = Q - S \cdot R^{-1} \cdot S^T \tag{6.17}$$

$$\tilde{A} = A - L \cdot V^{-1} \cdot C \tag{6.18}$$

$$\tilde{W} = W - L \cdot V^{-1} \cdot L^T. \tag{6.19}$$

The control policy of a soft-constrained linear-quadratic dynamic game is obtained by the dynamic programming approach. The solution of the dynamic programming is a back-

ward recursion based discrete-time generalized Riccati equation, which calculates the positive definite matrices $P_{N+1}^{C}, P_{N}^{C}, \ldots, P_{1}^{C}$

$$P_n^C = \bar{A}^T \cdot P_{n+1}^C \cdot \bar{A} - \bar{A}^T \cdot P_{n+1}^C \cdot \Lambda_{n+1}^C \cdot P_{n+1}^C \cdot \bar{A} + \bar{Q} \qquad P_{N+1}^C = X^C \tag{6.20}$$

where the intermediate variables Λ_{n+1}^{C} and Γ_{n+1}^{C} are defined in the following, and the initialization of the Riccati recursion is selected such that $X^C \geq 0$ is a positive definite, (See "The Riccati Recursion" on page 128.)

$$\Lambda_{n+1}^C = \begin{bmatrix} B & D \end{bmatrix} \cdot \Gamma_{n+1}^C \cdot \begin{bmatrix} B^T \\ D^T \end{bmatrix} \tag{6.21}$$

where the term Γ_{n+1}^{C} in the quadratic form, an indefinite matrix in the Riccati recursion, is defined as

$$\Gamma_{n+1}^C = \begin{bmatrix} R + B^T \cdot P_{n+1}^C \cdot B & B^T \cdot P_{n+1}^C \cdot D \\ D^T \cdot P_{n+1}^C \cdot B & -\gamma^2 \cdot I_l + D^T \cdot P_{n+1}^C \cdot D \end{bmatrix}^{-1}. \tag{6.22}$$

The unique feedback saddle-point solution exists if the solvability test for the Riccati recursion, in the form of positivity condition, is satisfied such that

$$\gamma^2 \cdot I_l - D^T \cdot P_{n+1}^C \cdot D > 0\,. \tag{6.23}$$

The positivity condition actually implies the nonsingularity of Γ_{n+1}^{C} inversion and also a positive definite P_n^C matrix calculation in (6.20) in the next recursion. If this condition is not met, then the game does not admit a saddle-point solution and the algorithm must be aborted and tested again with an increased γ.

The worst-case state estimator for a soft-constrained linear-quadratic dynamic game is obtained by the dynamic programming approach. The solution of the dynamic programming is a forward recursion based on a discrete-time generalized Riccati equation, which calculates the positive definite matrices $P_1^F, P_2^F, \ldots, P_N^F$ such that

$$P_{n+1}^F = \tilde{A} \cdot P_n^F \cdot \tilde{A}^T - \tilde{A} \cdot P_n^F \cdot \Lambda_n^F \cdot P_n^F \cdot \tilde{A}^T + \tilde{W} \qquad P_0^F = X^F \tag{6.24}$$

where the intermediate variables Λ_n^F and Γ_n^F are defined below and the initialization for the Riccati recursion is selected such that $X^F \geq 0$ is a positive definite (See "The Riccati Recursion" on page 128.)

$$\Lambda_n^F = \begin{bmatrix} C^T & H^T \end{bmatrix} \cdot \Gamma_n^F \cdot \begin{bmatrix} C \\ H \end{bmatrix} \tag{6.25}$$

where the term Γ_n^F in the quadratic form, an indefinite matrix in the Riccati recursion, is the inverse of the *innovation variance* defined as

$$\Gamma_n^F = \begin{bmatrix} V + C \cdot P_n^F \cdot C^T & C \cdot P_n^F \cdot H^T \\ H \cdot P_n^F \cdot C^T & -\gamma^2 \cdot I_p + H \cdot P_n^F \cdot H^T \end{bmatrix}^{-1}. \tag{6.26}$$

The algorithm admits a unique feedback saddle-point solution if the solvability test for the Riccati recursion, in the form of positivity condition

$$\gamma^2 \cdot I_p - H \cdot P_n^F \cdot H^T > 0 \tag{6.27}$$

is satisfied.

This condition actually implies the nonsingularity of the Γ_n^F inversion and a positive definite P_{n+1}^F matrix calculation in (6.24) in the next recursion. It can be shown that the existence of Γ_n^F is related to the question of how well the state $\hat{x}_n$ can be estimated on the basis of the observations of y. For instance, if $y = x$ it will be shown that $\Gamma_n^F = 0$ [78]. If this condition is not satisfied, the game does not admit a saddle-point solution and the algorithm must be aborted and tested again with increased γ.

By introducing the following two intermediate variables

$$\overline{P_n^C} = \overline{A}^T \cdot ((P_n^C)^{-1} - \gamma^{-2} \cdot W)^{-1} \cdot \overline{A} + \overline{Q} \tag{6.28}$$

and

$$\overline{P_{n+1}^F} = \tilde{A} \cdot ((P_n^F)^{-1} - \gamma^{-2} \cdot Q)^{-1} \cdot \tilde{A}^T + \tilde{W} \tag{6.29}$$

the coupling condition can be expressed in its usual simple form.

It is required that the coupling condition, expressed in the form of the positivity test, is satisfied by either

$$\gamma^2 \cdot I_m - P_n^F \cdot \overline{P_n^C} > 0 \tag{6.30}$$

or

$$\gamma^2 \cdot I_m - \overline{P_n^F} \cdot P_n^C > 0\,. \tag{6.31}$$

It is a kind of test whether state estimation and the state feedback combined in some way yield the desired result: an internally stabilizing feedback which makes the H_∞ norm

less than γ [78]. If the coupling condition is not satisfied, the selected γ is too small. Therefore, the algorithm must be aborted and tested again using an increased γ.

If all the positivity tests, (6.23), (6.27), and (6.30) or (6.31), are satisfied then the (sub)optimal state-feedback controller Σ_K can be written as

$$\mu^*(x_n) : \qquad u_n = K^x \cdot \hat{x}_n \tag{6.32}$$

where the state-feedback gain for the robust H_∞ (sub)optimal controller is defined as

$$K^x = -R^{-1} \cdot (B^T \cdot \Delta^C \cdot \overline{A} + S^T) \cdot \Delta^S \tag{6.33}$$

$$\Delta^C = ((P_n^C)^{-1} + B \cdot R^{-1} \cdot B^T - \gamma^{-2} \cdot W)^{-1} \tag{6.34}$$

$$\Delta^S = (I_m - \gamma^{-2} \cdot P_n^F \cdot P_n^C)^{-1}. \tag{6.35}$$

The robust state estimator can be written in the form

$$\hat{x}_{n+1} = F^x \cdot \hat{x}_n + F^u \cdot u_n + F^y \cdot y_n \tag{6.36}$$

where the matrices F^x, F^u, and F^y are defined as

$$F^x = A + \gamma^{-2} \cdot \tilde{A} \cdot \Delta^F \cdot Q - (\tilde{A} \cdot \Delta^F \cdot C^T + L) \cdot V^{-1} \cdot C \tag{6.37}$$

$$F^u = B + \gamma^{-2} \cdot \tilde{A} \cdot \Delta^F \cdot S \tag{6.38}$$

$$F^y = (\tilde{A} \cdot \Delta^F \cdot C^T + L) \cdot V^{-1} \tag{6.39}$$

and the intermediate matrix Δ^F is calculated as

$$\Delta^F = ((P_n^F)^{-1} + C^T \cdot V^{-1} \cdot C - \gamma^{-2} \cdot Q)^{-1}. \tag{6.40}$$

The H_∞ (sub)optimal controller, obtained by the equations (6.32) and (6.36), ensures an attenuation level of γ for the worst-case external disturbances, as explained in Chapter 4.

6.2.4 Step-by-Step Implementation

With the integration of sections 6.2.1, 6.2.2 and 6.2.3, the step-by-step implementation of the H_∞ (sub)optimal control algorithm is described. The efficient implementation of the controller requires the proper integration of the search for the infimum γ_0. The complete H_∞ (sub)optimal algorithm with the infimum search can be defined in the following:

1. The process starts with the selection of an initial value $\gamma > 0$. Go to step 2.
2. The infimum search with $\gamma_F = 0$ and $\gamma_P = 0$ is initialized. The equations from (6.14) through (6.19) are calculated. Go to step 4.
3. Defined by the inequality (6.23), the solvability test is performed. If the positive definiteness is satisfied go to step 4. If the solvability condition is not satisfied, the selected γ is too small. In this situation, go to step 10.
4. The backward Riccati recursion for the control is calculated using the equations (6.20) through (6.22). Go to step 5.
5. The solvability test, defined by the inequality (6.27), is performed. If the positive definiteness is satisfied go to step 6, else go to step 10.
6. The forward Riccati recursion for the state estimation is calculated using equations (6.24) to (6.26). Continue with step 7.

7. In preparation for testing the coupling condition, some intermediate matrices, defined by equations (6.28) and (6.29), are calculated. Go to step 8.
8. The coupling condition is evaluated using either (6.30) or (6.31) inequality. If the test is successful, γ becomes the upper limit in the search $\gamma_P = \gamma$. Go to step 9. If the test is unsuccessful go to step 10.
9. A valid $\gamma \in \Gamma_P$, which satisfies all the positivity tests, is confirmed. At this step, a decision is made whether the search for the infimum γ_0 should be terminated or not. If the inequality test is satisfied $\varepsilon > (\gamma_P - \gamma_F)/\gamma_P$, where ε is the selected tolerance, continue with step 11. Otherwise $\gamma = (\gamma_P + \gamma_F)/2$ and return to step 4.
10. The iteration for the selected γ is unsuccessful $\gamma \in \Gamma_F$; therefore, the lower limit on γ should be modified to $\gamma_F = \gamma$. Furthermore, if $\gamma_P \neq 0$ then $\gamma = (\gamma_P + \gamma_F)/2$ else $\gamma = 2 \cdot \gamma$. Go to step 4.
11. An infimum $\gamma_0 = \gamma_P$, within a desired accuracy, is obtained. For a stable solution, it is suggested in [78] that $\gamma = 1.1 \cdot \gamma_0$ should be used for control calculations. Go to step 12.
12. With the defined γ, the equations from (6.20) to (6.31) are recalculated. All the tests are expected to pass. Go to step 13.
13. To obtain the controller defined by equation (6.32), equations (6.33) to (6.35) are calculated for the control gain. Continue with step 14.
14. The state-space estimator, defined by equation (6.36), requires that the matrices are calculated using equations (6.37) to (6.40). Continue with step 15.

15. The measurement feedback H_∞ (sub)optimal control algorithm, defined by equations (6.32) and (6.36), is ready for real time implementation.

The algorithm, calculated through the above 15 steps, defines a central H_∞ (sub)optimal measurement feedback controller such that the closed-loop system mapping $\omega \to z$ has an induced norm less than γ, where γ is calculated in step 11.

6.2.5 The Riccati Recursion

The control system design for the time-invariant discrete-time plant with linear-quadratic objective function, typically results in the well studied *Discrete-time Algebraic Riccati Equation* (DARE). On the other hand, the control system design for the time-varying discrete-time plant with linear-quadratic objective function, uses the Riccati equation in its recursive form.

If the discrete-time plant is time-invariant the controller can be calculated either by the DARE or by the Riccati equation in its recursive form. It is a reasonable expectation that the control system design obtained using both these methods will result in an equivalent design. As shown in [20] and [7], the recursive Riccati equations (6.20) and (6.24) converge to the desired solution, equivalent to the solution of DARE, depending on the value chosen for (X^C and X^F). The initial value is required to be positive semidefinite $X^C \geq 0$ and $X^F \geq 0$. The convergence will not be guaranteed if the initialization matrices are too large.

In case of the Riccati recursion for controller calculation in equation (6.20), if the initial matrix X^C is selected such that

$$0 \le X^C \le \Pi^C \tag{6.41}$$

where Π^C is the solution of the Lyapunov equation

$$\Pi^C = \bar{A}^T \cdot \Pi^C \cdot \bar{A} + \bar{Q} \tag{6.42}$$

and where $\bar{A}$ is defined by (6.16) and $\bar{Q}$ is defined by (6.17), the Riccati recursion sequence $P_{N+1}^C, P_N^C, \ldots, P_1^C$ will converge to the unique stabilizing solution of an equivalent DARE, if it exists. Consequently, the sequence $P_{N+1}^C, P_N^C, \ldots, P_1^C$ is also bounded by $0 \le P_n^C \le \Pi^C$, otherwise the initial condition will be violated.

In case of the Riccati recursion for state-space estimation in equation (6.24), if the initial matrix X^F is selected such that

$$0 \le X^F \le \Pi^F \tag{6.43}$$

where Π^F is the solution of the Lyapunov equation

$$\Pi^F = \tilde{A} \cdot \Pi^F \cdot \tilde{A}^T + \tilde{W} \tag{6.44}$$

and where $\tilde{A}$ is defined by (6.18) and $\tilde{W}$ is defined by (6.19), the Riccati recursion sequence $P_1^F, P_2^F, \ldots, P_N^F$ will converge to the unique stabilizing solution of an equivalent DARE, if it exists. Consequently, the sequence $P_1^F, P_2^F, \ldots, P_N^F$ is also bounded by $0 \le P_n^F \le \Pi^F$, otherwise the initial condition will be violated.

6.3 Implementation Example

To illustrate the implementation of the measurement feedback H_∞ (sub)optimal control algorithm, an example is selected involving the mass spring system. Originally proposed in [103], known as the ACC benchmark problem from the American Control Conference, this example is widely used in the control literature [103-105].

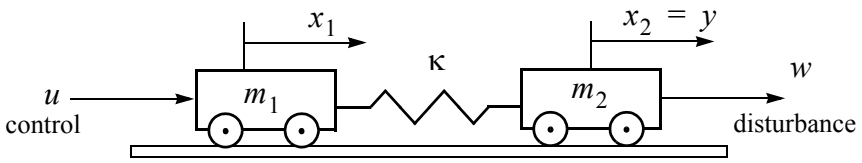


Figure 6-4 ACC benchmark problem

The plant consists of two masses m_1 and m_2, connected with the spring of stiffness κ, and the plant damping is expressed by ζ. The challenge for the control design is originating from the very large phase lags in the control system due to the non collocated actuator/ sensor set-up. The sensor y senses the position of the mass m_2 while the actuator u acts on the mass m_1. The state-space representation of the plant model in continuos time is given by the equations

$$P: \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\kappa/m_1 & \kappa/m_1 & -\zeta/m_1 & \zeta/m_1 \\ \kappa/m_2 & -\kappa/m_2 & \zeta/m_2 & -\zeta/m_2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix} \cdot u \qquad (6.45)$$

$$y = x_2$$

where $x_1, \ldots, x_4$ are the states of the system: x_1 is the position of the mass m_1, x_2 is the position of the mass m_2, $x_3 = \frac{dx_1}{dt}$ is the speed of the mass m_1, $x_4 = \frac{dx_2}{dt}$ is the speed of the mass m_2, u is the control input, and y is the plant output.

The design requirement is to find a controller that stabilizes the system with an uncertain spring constant κ, varying between *0.5* and *2*, with the nominal value set to *1*. Both masses are selected to be equivalent $m_1 = m_2 = 1$. The plant damping is set for $\zeta = 0.01$. However, even if $\zeta = 0$, i.e., the plant is undamped, simulation studies show that the control system performance is not significantly affected by this value.

The open loop plant response for an initial disturbance in the state $x_{1,\,t=0} = 1$ is presented in Figure 6-5. This disturbance is used to test and demonstrate the behavior of the control system. The discrete time representation, P_d, of the continuous time system, P, from (6.45), is obtained by substituting the nominal parameter values and using the matlab function c2d with the sampling time of $T_S = 0.05s$ and the zero-order-hold (ZOH) option

P_d:

$$x_{n+1} = \begin{bmatrix} 0.99875 & 0.0012491 & 0.049967 & 3.3314e{-}5 \\ 0.0012491 & 0.99875 & 3.3314e{-}5 & 0.049967 \\ -0.049933 & 0.049933 & 0.99825 & 0.0017484 \\ 0.049933 & -0.049933 & 0.0017484 & 0.99825 \end{bmatrix} \cdot x_n + \begin{bmatrix} 0.0012495 \\ 4.6855e{-}7 \\ 0.049967 \\ 3.3314e{-}5 \end{bmatrix} \cdot u \quad (6.46)$$

$$y_n = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \cdot x_n$$

In order to compare the discrete time representation, P_d, with the original continuous time system, P, the open loop plant response for an initial disturbance in the state $x_{1,\,t=0} = 1$ of P_d is also plotted in Figure 6-5.

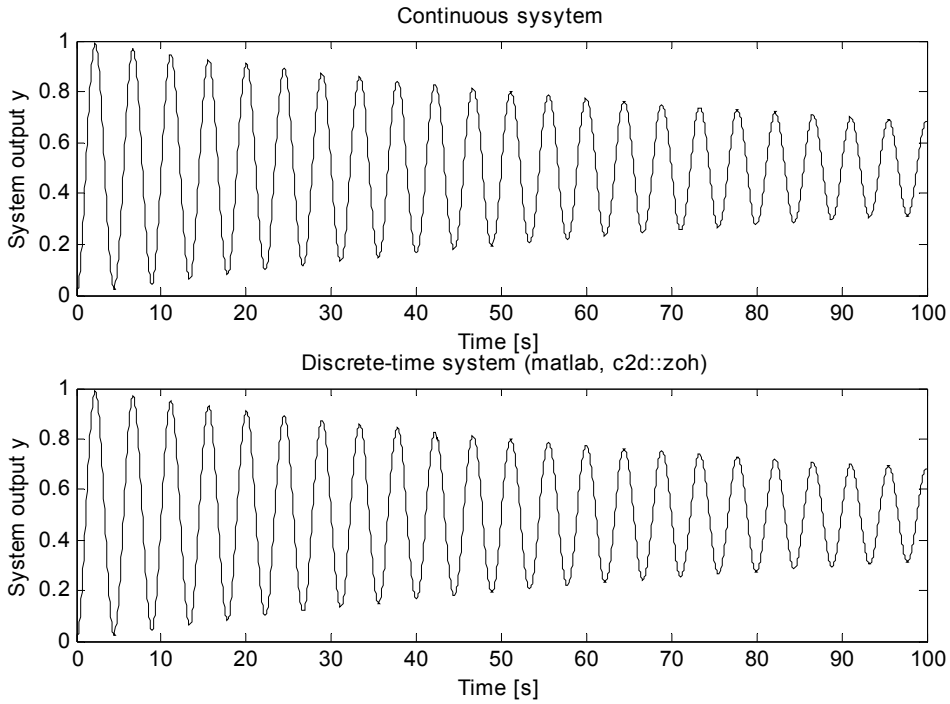


Figure 6-5 Open loop system response with nominal parameters

To formulate the robustness and performance requirements for the standard H_∞ control algorithm in the form of (6.1), it is required to choose the appropriate matrices $D \in \Re^{q \times m}$, $E \in \Re^{q \times l}$, $H \in \Re^{p \times m}$ and $G \in \Re^{p \times r}$. For the selection of these matrices, in the case of single-input/single-output (SISO) system design, graphical tools, like the Bode and Nyquist diagrams, can be used. For multiple-input/multiple-output (MIMO) systems the selection of these matrices is not straightforward [78]. One possible solution for MIMO system design would be the implementation of the trial and error approach.

In this research, the trial and error approach is applied to select the proper weighting matrices by observing the plant behavior through its output y_n. In the following, this procedure is illustrated.

Through the selection of matrices $H \in \Re^{p \times m}$ and $G \in \Re^{p \times r}$, the weights for the performance index are determined. The dimensions m and r are defined by the plant model. The dimension p, which can be selected freely, defines the size of the controlled output $z \in \Re^p$ vector. One good choice for the initial guess, is to select that $p = m + r$. In this case,

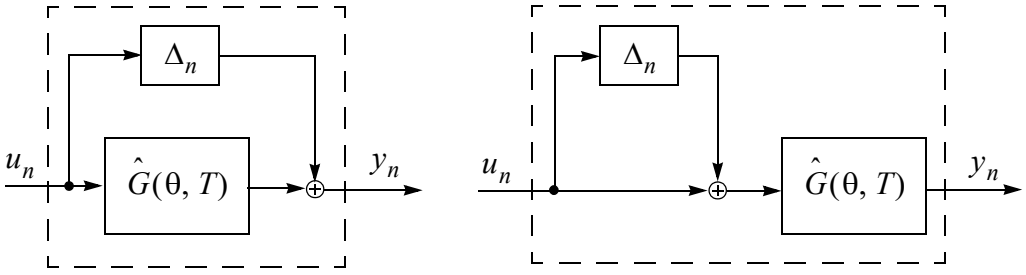
$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{6.47}$$

then the controlled output contains all the state and control variables

$$z_n = \begin{bmatrix} x_{1,n} \\ x_{2,n} \\ x_{3,n} \\ x_{4,n} \\ u_n \end{bmatrix}. \tag{6.48}$$

The selection of $D \in \Re^{q \times m}$ and $E \in \Re^{q \times l}$ is more complex because through these matrices the model uncertainty and the effects of the exogenous disturbances for the control system are modelled.

For better understanding of the functions of the D and E matrices, Figure 6-6 shows the modelling of the disturbances as additive perturbations and multiplicative perturbations. Matrix D is used to interpret the model perturbations effecting the states of the plant model, and matrix E is used to interpret the disturbances equivalent to the measurement noise.



$$x_{n+1} = A \cdot x_n + B \cdot u \qquad x_{n+1} = A \cdot x_n + B \cdot u + B \cdot \Delta_n$$

$$y_n = C \cdot x_n + \Delta_n \cdot u \qquad y_n = C \cdot x_n + \Delta_n \cdot u$$

Figure 6-6 Additive and multiplicative perturbations

The m and l dimensions for the matrices $D \in \Re^{q \times m}$ and $E \in \Re^{q \times l}$ are defined in the plant model. The dimension q shall be selected according to the requirement to adequately represent the perturbations effecting the plant model.

Testing of the control system design with the dimension $q = 1$, always resulted in an unstable controller. Therefore it is concluded that the dimension of q can not be selected freely, as its lower value is limited. In the first and second trial, as presented below, $q = 2$ and $q = 4$ were selected respectively, resulting in a stable control system.

6.3.1 First Tryout

In the first tryout, it is expected that the disturbances on the masses m_1 and m_2 have the same accelerating force as the control signal. This is the reason for selecting the values for the matrix D similarly to the matrix B, in the first tryout

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1/m_1 \\ 1/m_2 & 0 \end{bmatrix}. \tag{6.49}$$

Substituting the nominal parameter values and by using the matlab function c2d with $T_S = 0.05s$ and with the zero-order-hold (ZOH) options, the discrete time matrix is defined as

$$D = \begin{bmatrix} 4.6855e{-}7 & 0.0012495 \\ 0.0012495 & 4.6855e{-}7 \\ 3.3314e{-}5 & 0.049967 \\ 0.049967 & 3.3314e{-}5 \end{bmatrix}. \tag{6.50}$$

During the first trial matrix D remains unchanged. Only the values in E, H and G matrices are manipulated. In the following the results from several tests are presented, which illustrate the effects of the selection of matrices E, H and G on the resulting control performance, through the control signal and the plant response.

Overall, thirteen test results are recorded, named as *Test a* through *Test m*. Since presenting all these tests in one graph would not result in any meaningful information, the visualization of tests is divided in four groups, named *Test Group I* through *Test Group IV.* Every graph consists of two parts. The upper part is the performance index obtained as a square sum of the plant output, over the time period of $100s$ from the disturbance. The lower part represents the control energy attained as a square sum of the control signal, over the same period of time. In each test the H_∞ controller is calculated when the nominal spring stiffness $\kappa = 1$ is substituted into the plant model. Once the controller is obtained, the spring stiffness in the plant model is varied from 0.2 up to 4 by 0.1 steps, and for all steps the control performance was recorded and visually presented.

Test a: $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = \begin{bmatrix} 1 & 1 \end{bmatrix}$

$$\text{Test b:} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad E = 0.1 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$\text{Test c:} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad E = 0.01 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$$

$$\text{Test d:} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

In *Test Group I,* Figure 6-7 compares the plants' performance indexes and the control energy as a function of spring stiffness, obtained from *Tests a, b, c,* and *d*. It is apparent that in this test group the controller in *Test a* is stable only for the nominal plant, while its best performance is demonstrated in *Test d*.

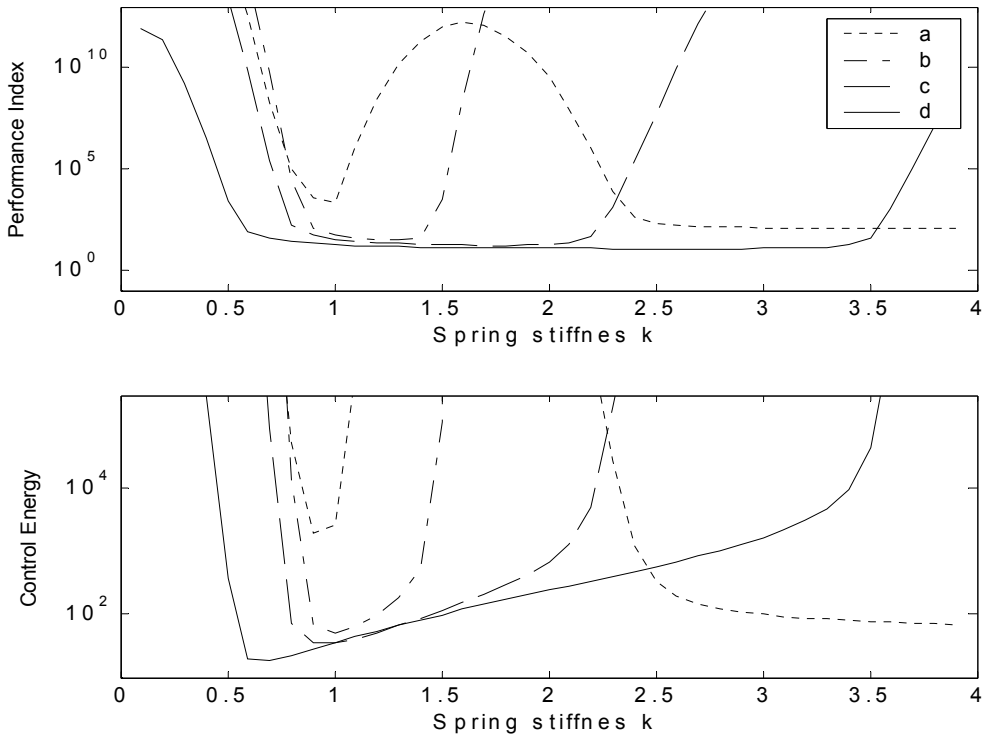


Figure 6-7 First tryout: Test Group I

Test e: $H = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$

Test f: $H = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$

Test g: $H = \begin{bmatrix} 300 & 0 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$.

In *Test Group II,* Figure 6-8 compares the plants' performance indexes and the control energy as a function of spring stiffness, obtained from *Tests e, f,* and *g*. To facilitate the

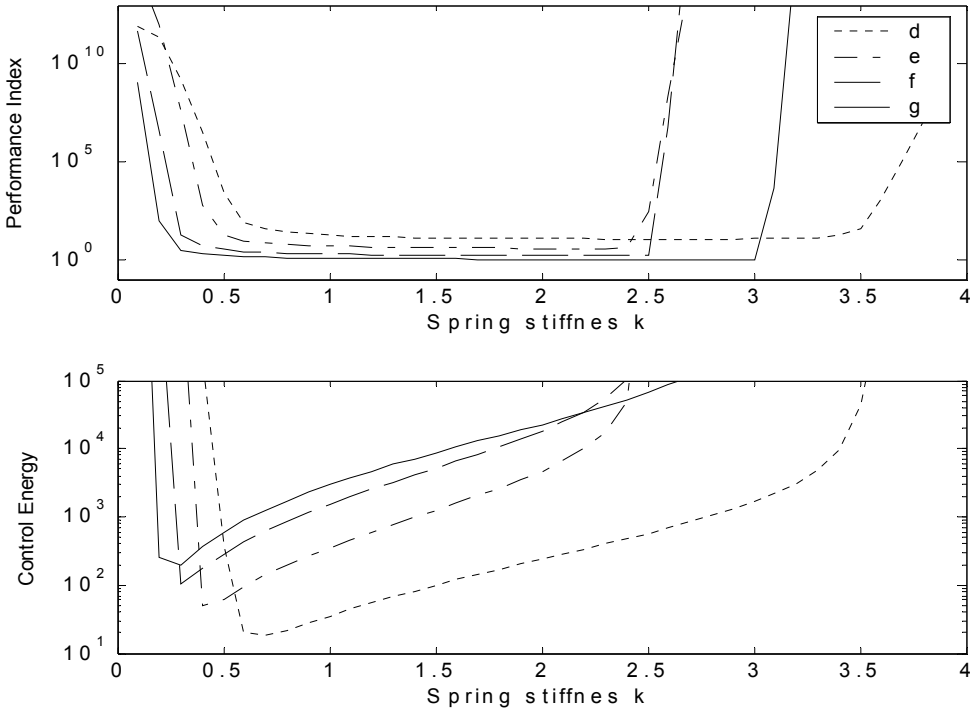


Figure 6-8 First tryout: Test Group II

comparison between the tests, *Test d* from *Test Group I* is also included. This test group shows significant improvement in performance, when dealing with lower values of spring stiffness. At the same time, the controller's performance deteriorates for higher values of spring stiffness. However, if it is taken into account that the control design solely requires

a stabilizing controller for spring stiffness between 0.5 and 2, all controllers belonging into this test group are admissible. It can be noted that the controller in *Test g* has the best performance.

Test h: $H = \begin{bmatrix} 300 & 0 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 5 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$

Test i: $H = \begin{bmatrix} 300 & 0 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 50 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$

Test j: $H = \begin{bmatrix} 300 & 0 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 300 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$.

For *Test Group III,* Figure 6-9 compares the plants' performance indexes and the control energy as a function of spring stiffness, obtained from *Tests h, i,* and *j*. To facilitate the comparison between the tests, *Test g* from *Test Group II* is also included. Compared to the performance of the controller in *Test g,* an overall performance degradation can be observed. At the same time, the control energy consumption is reduced. The controller designed in *Test h,* satisfies all the control design requirements.

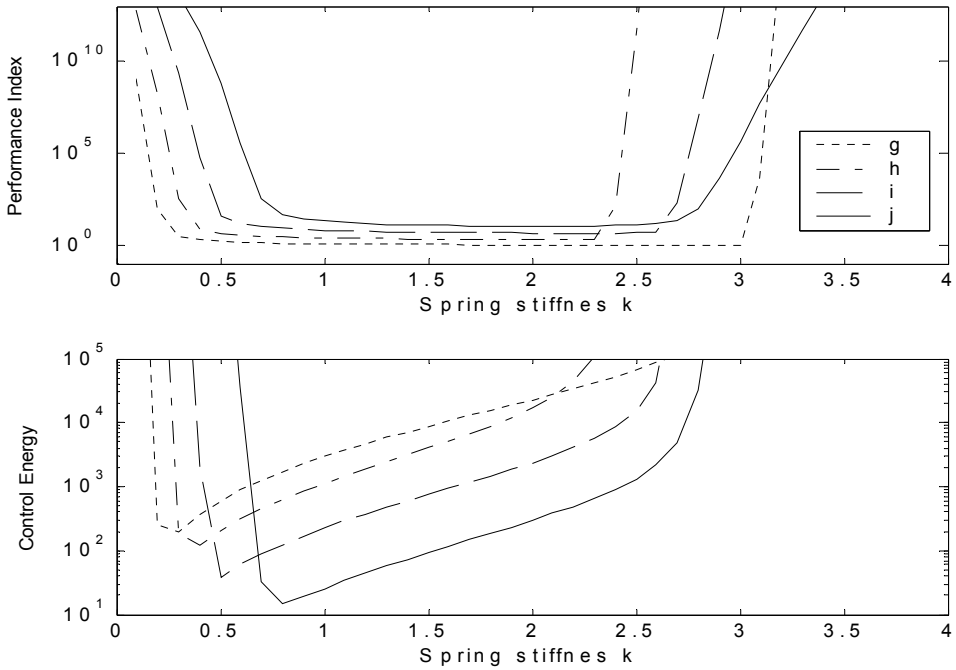


Figure 6-9 First tryout: Test Group III

Test k: $H = \begin{bmatrix} 300 & 0 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.8 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$

Test l: $H = \begin{bmatrix} 300 & 0 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.6 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$

Test m: $H = \begin{bmatrix} 300 & 0 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.5 \end{bmatrix}$ $E = 0.003 \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}$.

For *Test Group IV,* Figure 6-10 compares the plants' performance indexes and the control energy as a function of spring stiffness, obtained from *Tests k, l,* and *m*. To facilitate

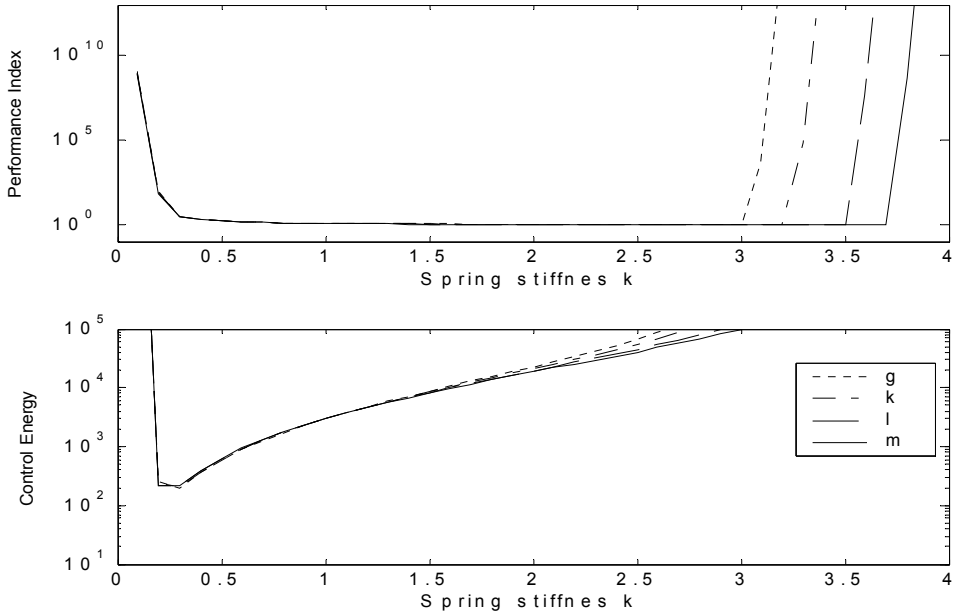


Figure 6-10 First tryout: Test Group IV

the comparison between the tests, *Test g* from *Test Group II* is included. Compared to the performance of the controller in *Test g,* the performance of the controllers and the control energy consumption are similar to each other. Although there are performance variations for higher values of spring stiffness, these performances are beyond the control design requirements.

From the engineering design perspective, where a compromise must be made between the control energy and controller performance, the design specified in *Test h* shall be selected for control implementation. For the resulting controller $\gamma = 330$, the matrices for the real-time control (6.2) are

$$F^x = \begin{bmatrix} 1.0585 & -31.501 & 0.049967 & 3.3368e{-}5 \\ 0.0018572 & -0.91708 & 3.3314e{-}5 & 0.049967 \\ -0.039576 & 5.6301 & 0.99825 & 0.0017484 \\ 0.05471 & -10.048 & 0.0017484 & 0.99825 \end{bmatrix} \qquad F^u = \begin{bmatrix} 0.0012495 \\ 4.6855e{-}7 \\ 0.049967 \\ 3.3314e{-}5 \end{bmatrix}. \quad (6.51)$$

$$F^y = \begin{bmatrix} 31.503 \\ 1.9158 \\ -5.5804 \\ 9.9985 \end{bmatrix} \qquad K^x = \begin{bmatrix} -108.25 & -36.187 & -15.103 & -192.47 \end{bmatrix}$$

6.3.2 Second Tryout

In the second tryout the effects of a more complex disturbance model are tested. In this model the length of four has been selected for the external disturbances vector ω. The appropriate noise model is defined as

$$D = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1/m_1 & 1/m_1 & 1/m_1 \\ 1/m_2 & 0 & -1/m_2 & 1/m_2 \end{bmatrix}. \quad (6.52)$$

By substituting the nominal parameter values and using the matlab function c2d with $T_S = 0.05s$ and the zero-order-hold (ZOH) option, the discrete time matrix is

$$D = \begin{bmatrix} 4.6855e{-}7 & 0.0012495 & 0.0012491 & 0.00125 \\ 0.0012495 & 4.6855e{-}7 & 0.0012491 & 0.00125 \\ 3.3314e{-}5 & 0.049967 & 0.049933 & 0.05 \\ 0.049967 & 3.3314e{-}5 & 0.049933 & 0.05 \end{bmatrix}. \tag{6.53}$$

During the trial, matrix D remains unchanged. Only the values in E, H and G matrices are manipulated. The description of the tests is equivalent to that presented in the first tryout, not repeated here. The test results are presented in the following

Test a: $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$

Test b: $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.1 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$

Test c: $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.01 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$

Test d: $H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$.

Like in the first tryout, in *Test Group I,* Figure 6-11 compares the plants' performance indexes and the control energy as a function of spring stiffness, obtained from *Tests a, b, c,* and *d*. It is apparent that in this test group the controller in *Test a* is stable only for the nominal plant, while its best performance is demonstrated in *Test d.*

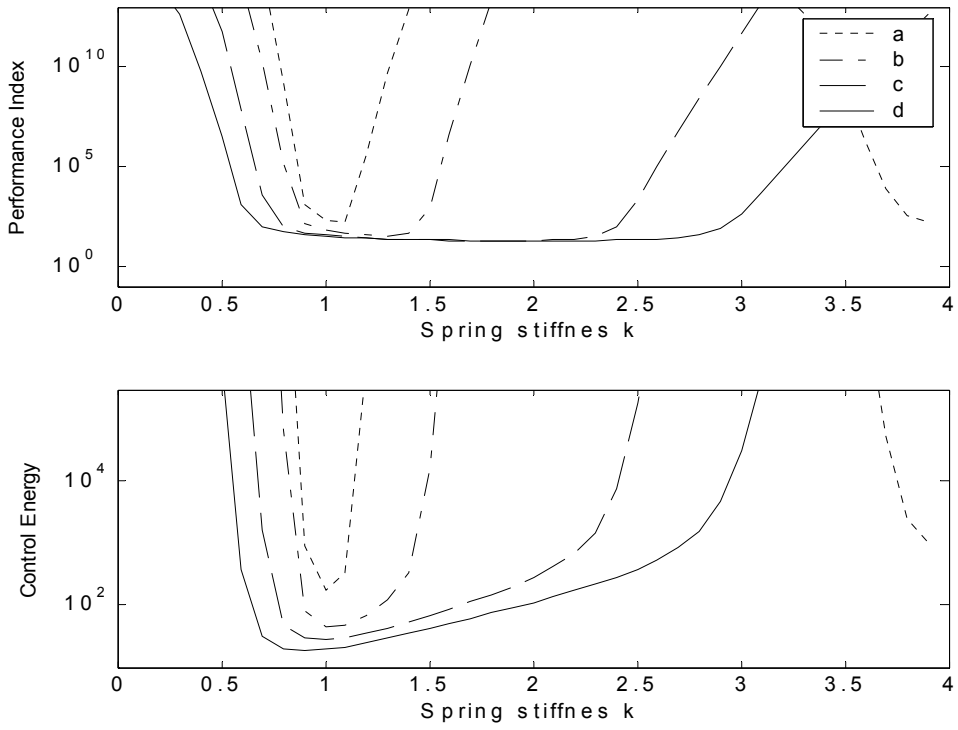


Figure 6-11 Second tryout: Test Group I

Test e: $H = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$

Test f: $H = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$

Test g: $H = \begin{bmatrix} 140 & 0 & 0 & 0 & 0 \\ 0 & 140 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$ $E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$.

In *Test Group II,* Figure 6-12 compares the plants' performance indexes and the control energy as a function of spring stiffness, obtained from *Tests e, f,* and *g*. To facilitate the comparison between the tests, *Test d* from *Test Group I* is also included. This test group shows some improvement in performance, when dealing with lower values of spring stiffness. As in the first tryout, the controller in *Tests f* and *g* has the best performance.

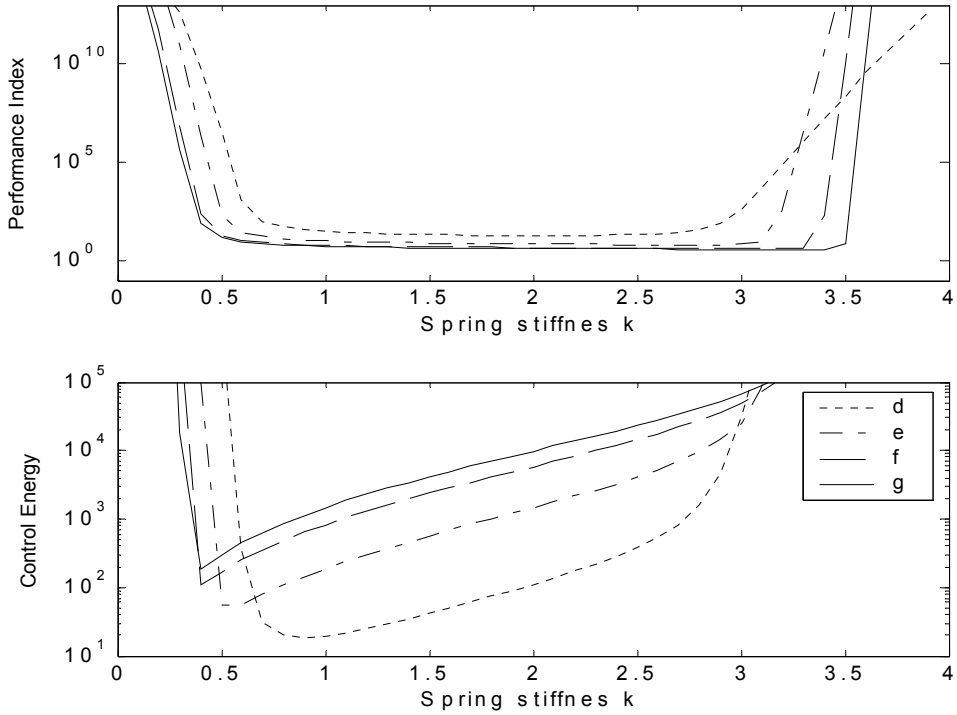


Figure 6-12 Second tryout: Test Group II

Test h: $H = \begin{bmatrix} 140 & 0 & 0 & 0 & 0 \\ 0 & 140 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}$ $E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$

Test i: $H = \begin{bmatrix} 140 & 0 & 0 & 0 & 0 \\ 0 & 140 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 100 \end{bmatrix}$ $E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$

Test j: $H = \begin{bmatrix} 140 & 0 & 0 & 0 & 0 \\ 0 & 140 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 150 \end{bmatrix}$ $E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$.

For *Test Group III,* Figure 6-13 compares the plants' performance indexes and the control energy as a function of spring stiffness, obtained from *Tests h, i,* and *j*. To facilitate the comparison between the tests, *Test g* from *Test Group II* is also included. Compared to the performance of the controller in *Test g,* an overall performance degradation can be observed, yet the controller designed in *Test h,* satisfies all the control design requirements.

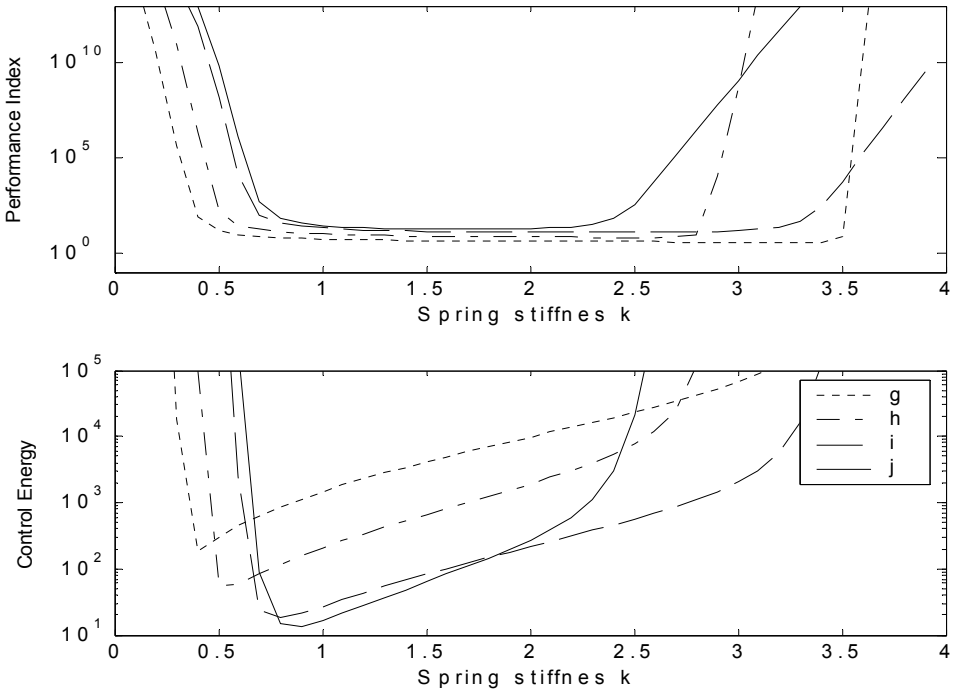


Figure 6-13 Second tryout: Test Group III

$$\text{Test k:} \quad H = \begin{bmatrix} 140 & 0 & 0 & 0 & 0 \\ 0 & 140 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.9 \end{bmatrix} \quad E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\text{Test l:} \quad H = \begin{bmatrix} 140 & 0 & 0 & 0 & 0 \\ 0 & 140 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.8 \end{bmatrix} \quad E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\text{Test m:} \quad H = \begin{bmatrix} 140 & 0 & 0 & 0 & 0 \\ 0 & 140 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.7 \end{bmatrix} \quad E = 0.005 \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}.$$

In *Test Group IV,* Figure 6-14 compares the plants' performance indexes and the control energy as a function of spring stiffness, obtained from *Tests k, l,* and *m*. To facilitate the comparison between the tests, *Test g* from *Test Group II* is included. Compared to the performance of the controller in *Test g,* the performance of the controllers and the control energy consumption are similar to each other. Although there are performance variations for higher values of spring stiffness, these performances are beyond the control design requirements.

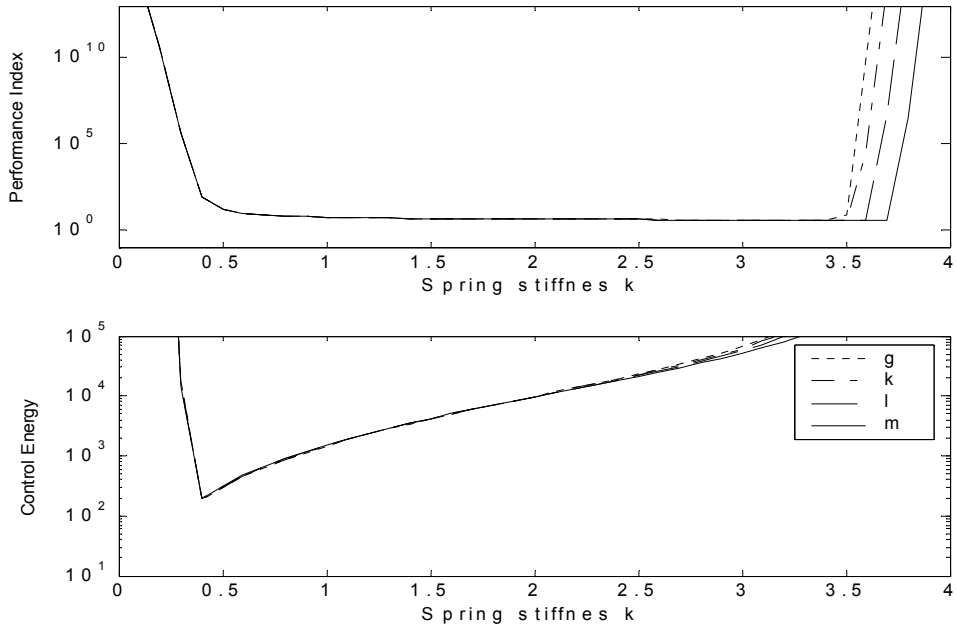


Figure 6-14 Second tryout: Test Group IV

From the engineering design perspective, where a compromise must be made between control energy and controller performance, the design specified in *Test h* shall be selected for control implementation. For the resulting controller $\gamma = 374$, the same as in the first tryout, the matrices for the real-time control (6.2), are

$$F^x = \begin{bmatrix} 1.029 & -14.02 & 0.049967 & 3.335e{-}5 \\ 0.001714 & -1.0221 & 3.3312e{-}5 & 0.049967 \\ -0.045472 & 8.3879 & 0.99825 & 0.0017483 \\ 0.054438 & -11.907 & 0.0017485 & 0.99825 \end{bmatrix} \qquad F^u = \begin{bmatrix} 0.0012495 \\ 4.6855e{-}7 \\ 0.049967 \\ 3.3314e{-}5 \end{bmatrix}. \quad (6.54)$$

$$F^y = \begin{bmatrix} 14.022 \\ 2.0209 \\ -8.3382 \\ 11.857 \end{bmatrix} \qquad K^x = \begin{bmatrix} -33.423 & -3.9344 & -8.8928 & -35.36 \end{bmatrix}$$

The resulting controller performance is very similar to that of the first tryout, even though the noise model in the second tryout is much more complicated. It can be concluded that the noise model, the dimension of the external disturbance vector ω, requires a specific minimum dimension. Therefore, in this example, no admissible controller can be obtained for the dimension of one. If the necessary dimension of the external disturbances vector ω is used, an admissible controller is obtained. Any further increase of the noise model does not significantly improve the controller's behavior. Either way, the controller's dimension is determined by the dimension of the plant model.

7 Model Predictive Control

The control system design, defined in the previous chapters, assumes the knowledge, with some accuracy, of the plant model for a time invariant system, or the plant model as a function of time for a time varying system. This assumption is rarely valid outside theoretical control research, especially in the design of controllers for industrial systems. In most cases, from observations of the plant's response to the input, it can be assumed that the system model is known up to the present time; however, the model validity is uncertain in the future. Theoretically, it is possible to construct models which describe the plant's future behavior but this would require a design effort which, in most industrial environments, is not economically feasible.

The so called *model predictive control* (MPC) is one of the few suitable methods, which can be used to resolve the controller design without the enormous task of identifying the exact (very accurate non-linear) plant model. This method describes the plant not only in the past but also in the near future and it can be used to control the changing plant.

Model predictive control (MPC), receding-horizon control (RHC), moving-horizon control [106], and generalized predictive control (GPC) are names in the literature used as synonyms to describe the same form of control. In this type of control, the current control action is determined by solving an on-line optimal control problem for every time interval [107-110].

When using MPC, the optimal control problem is required to have a finite horizon. The resulting control law is a feedback control, usually calculated using state models, where the control u_n* is dependent on the plant's state x_n [106].

The optimal control problem, when calculated at each sampling time, is referred to as the *pointwise strategy,* which is presented in Figure 7-1. The terminal point in this control is of fixed-length, $N \cdot T_s$ (T_s is the sampling interval), where the finite cost horizon continuously recedes at each time instant [111]. The implementation of the pointwise strategy requires processing power which is capable of calculating the optimal strategy in real time. This calculation must be performed in less time than the sampling interval T_s.

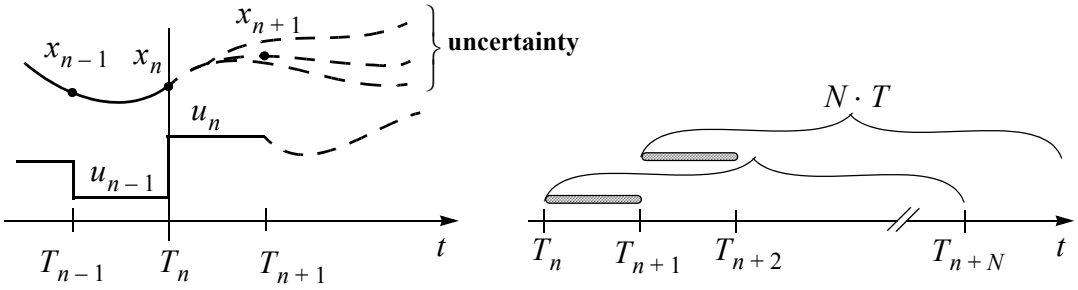


Figure 7-1 Model predictive control strategy - pointwise

In most industrial control systems, the processing power presently available is not sufficient for the implementation of the H_∞ (sub)optimal control algorithm. In ten to fifteen years, there will probably be computers on the market with sufficient processing power to allow the implementation of the pointwise strategy. Until then, the so called intervalwise strategy should be used for implementation.

In the *intervalwise strategy,* the terminal point $N \cdot T_s$ is kept fixed for the finite cost horizon and after an interval of $N_c \cdot T_s$, the terminal point moves by $N_c \cdot T_s$ and the strategy is fixed for the next $N_c \cdot T_s$ period, as presented in Figure 7-2.

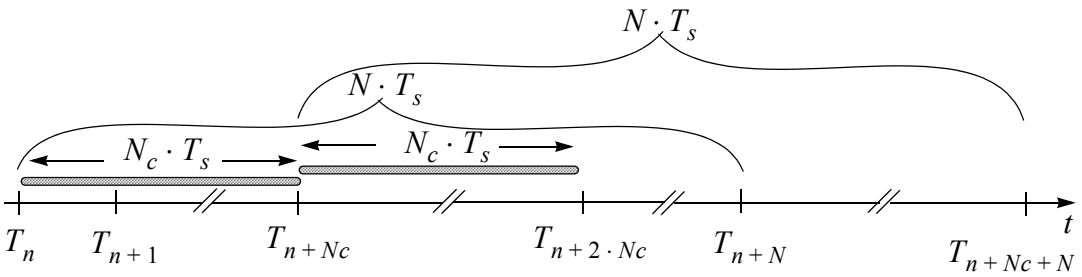


Figure 7-2 Model predictive control strategy - intervalwise

The intervalwise receding-horizon technique, at each time $t = N_c \cdot T_s$, solves an optimization problem based on a cost function evaluated on the interval $[t, t + N \cdot T_s]$, for some fixed $N \cdot T_s$. The resulting controller is implemented on a shorter interval, say $[t, t + Nc \cdot T_s]$. Then, at time $t + Nc \cdot T_s$ the optimization is performed again, with the cost function applied to the interval $[t + Nc \cdot T_s, t + (N + Nc) \cdot T_s]$. For the selection of the interval $N_c \cdot T_s$, it is required that this interval be smaller than the selected finite horizon interval for the control policy $N_c < N$ and that this interval be longer than the required time for all the calculations, as illustrated in Figure 7-3. The controller calculations entail the following:

1. system parameter estimation t_{ce},

2. model selection t_{cm},

3. control policy calculation t_{cp},

4. smoothing update of the state space observer t_{cs},

5. real time feedback control calculation t_{cr}.

In this technique, the calculations for the system parameter estimation t_{ce} and the real time feedback control t_{cr} are performed in real time. Therefore, the minimum requirements for the $N_c \cdot T_s$ interval can be defined as

$$\begin{gathered} t_{ce} + t_{cm} + t_{cp} + t_{cs} + t_{cr} < N_c \cdot T_s \\ \text{and} \\ t_{ce} + t_{cr} < T_s \end{gathered} \quad . \tag{7.1}$$

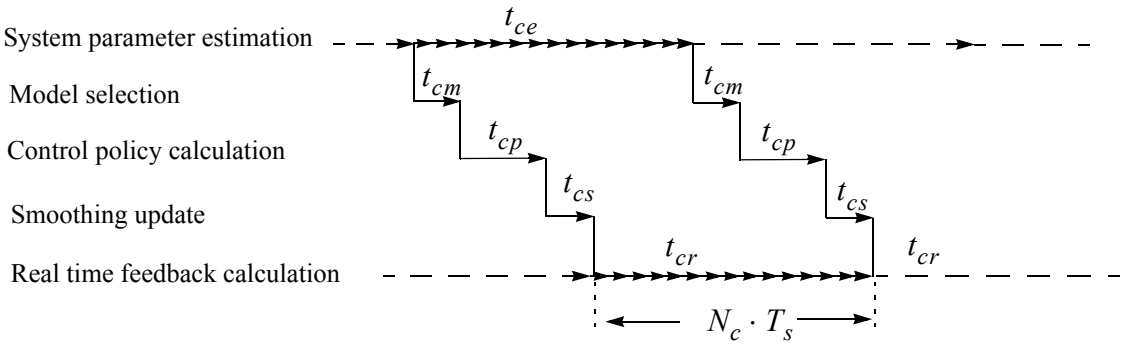


Figure 7-3 MPC as a multi-tasking function

If the requirements for the MPC implementation, beside the granted stability, also include the induced l_2 norm bound from disturbances to states, the implementation of a robust control algorithm, usually in the form of H_∞ (sub)optimal control, is needed. The

time domain game theoretic technique allows the solution of finite horizon problems with non-zero initial conditions and therefore supports the MPC formulation [112]. The constructed MPC system is affected by external disturbances, which raises the question of the MPC's stability under these perturbations. In the following, the requirements of the MPC's stability are defined for real-time stability analysis.

7.1 Requirements for Stability

The feedback controller, resulting from the application of the MPC strategy, is a stabilizing controller as long as the design parameters satisfy the MPC stability conditions below [106] [107].

The *terminal constraint set* X_f is a subset of X which contains the neighborhood of the origin. The task of the MPC is to steer the state of the plant into X_f in a finite time which is less than the selected control horizon time $t = N \cdot T_s$. In other words, it is required that $x_{n+N} \in X_f$ for every initial state $x_n \in X$ which is automatically satisfied for a controllable system, if N is chosen sufficiently large. The terminal constraint set X_f is selected to be the maximal admissible output set for the system $x_{n+1} = f_n(x_n, u_n)$. It is required that the *admissible MPC sequence* $\{u_n, \ldots, u_{n+N}\} \in U$, obtained by solving the optimal control problem, steers every initial state $x_n \in X$ of the plant into the terminal constraint set $x_{n+N} \in X_f$ in finite time $t = N \cdot T_s$. Further, it is expected that the admissible control sequence $\{u_{n+N+1}, \ldots\} \in U$ satisfies the control constraint when the plant state belongs

to the terminal constraint set $x_{n+N} \in X_f$. Furthermore, it is required that if the state of the plant belongs to the terminal constraint set $x_{n+N} \in X_f$ the future admissible control sequence $\{u_{n+N+1}, \ldots\} \in U$ acts in a way so that the plant state stays in the terminal constraint set $\{x_{n+N+1}, \ldots\} \in X_f$ which concurs that X_f *is positively invariant*[1] *under the admissible MPC sequence*.

The *stability of the MPC* technique can be established by using the Lyapunov stability criterion, which requires that the Lyapunov function defined on the MPC strategy be non-positive. As suggested in [107], the value function, $V_n(x_n)$ defined by (5.13), is widely accepted as a Lyapunov function in MPC. Now, if one deals with a "practical" system that is arising out of physics or technology, it is quite certain that perturbations will come up not only because of initial conditions other than $x_0 = 0$ but also, and above all, due to the action of exterior and adventitious actions, for example fairly brusque impulses [50]. As a consequence, the strict Lyapunov stability theory cannot be expected to validate the stability of MPC, rather the *practical stability* should be used. Following the same logic as defined in the section "Practical Stability" on page 34 and by using the definition introduced for the continuous systems in [50], the stability requirements for MPC can be defined as below.

---

1. A set X is called *positively invariant* if $\phi(X) \subset X$. It is called *invariant* if $\phi(X) = X$. For a dynamic system, a subset of the state space is said *invariant* if the state is in the set at some time, then it is in the set in the past and in the future. A subset of the state space is said to be *positively invariant* if it contains the system state at some time, then it will stay in it at the future time [108].

Theorem Let $V_n(x_n)$ be a Lyapunov function for the MPC system, a scalar function which for all $\{x_n\}$ has discrete difference and with the property that $V_n(x_n) \to \infty$ as $\|x_n\| \to \infty$. Let $\nabla V_n(x_n)$ denote the discrete time derivative of the Lyapunov function. If $\nabla V_n(x_n) \le -\varepsilon < 0$ for all x_n outside X_f for all ω_n in W, and all $\{n, n+1, \ldots\}$ and if $V_n(\underline{x_n}) \le V_n(\overline{x_n})$ for all $\underline{x_n}$ in X_f and all $\overline{x_n}$ in X_e, then the system possesses a *strong practical stability*. Moreover, it is required that the state of the system outside X_f be for a finite time T_e or less. The definition of ε in practical stability could then be limited by requiring that if $V_n(x_n) \le v_1$ in X_f and $V_n(x_n) \ge v_2$ in X_e then if $\varepsilon = (v_2 - v_1)/T_e$ implies that the plant state for the stable system outside X_f will remain for the time T_e or less.

In the determination of the practical stability the boundedness of the disturbances $\omega_n \in W$ defined by γ, as introduced in (4.5) during the development of the H_∞ (sub)optimal controller, is a necessary condition. For better understanding, this stability definition is illustrated as a trajectory of the Lyapunov function, in Figure 7-4. In this figure the

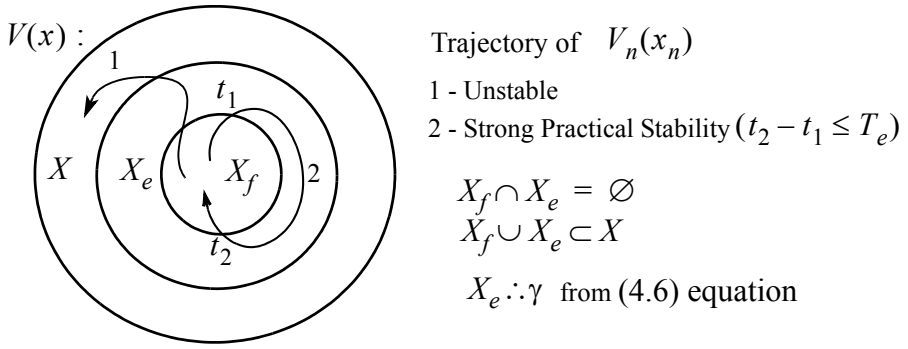


Figure 7-4 Practical stability

region characterized by the X_e set, corresponds to the uncertainty space γ defined by the (4.6) equation limit which is calculated as the H_∞ norm of the control system.

The sufficient stability requirements for an MPC can be summarized as in [107] and extended by the practical stability from [50]:

1. $X_f \subset X, X_f$ closed, $0 \in X_f$ (state constraint satisfied in X_f),
2. $(u_n = \mu(x_n)) \in U, \forall x_n \in X_f$ (control constraint satisfied in X_f),
3. $(x_{n+1} = f_n(x_n, u_n, \omega_n)) \in X_f, \forall x_n \in X_f, \forall \omega_n \in W$ (X_f is positively invariant under $u_n = \mu(x_n)$),
4. $\nabla V_n(x_n, \omega_n) \leq \varepsilon, \forall x_n \in X_e, \forall \omega_n \in W,$ (is a local Lyapunov function).

The stability theory introduced in this section is not limited by any assumption of the system linearity. On the other hand, the control algorithm defined in Chapter 6, is limited to linear systems. In the following, the implementation of this linear H_∞ control algorithm by the MPC technique for non-linear systems is investigated.

7.2 Robust MPC for Non-Linear Systems

Although the theory for non-linear H_∞ control is well developed, there have been very few practical applications due to the lack of efficient numerical methods for solving these problems. To solve the non-linear state-feedback H_∞ control problem, the solution of the Hamilton-Jacobi-Isaacs equation (inequality) needs to be attained [87] [113]. Since this

equation is a first order partial differential equation, solving it in real-time with the current theoretical framework is an impossible computational task.

Studies show that beside its advantages, the implementation of a non-linear controller has some disadvantages. In general, the region of attraction of the non-linear controller is larger than that of the linear controller. However, one should be cautioned to the fact that due to the higher-order polynomial terms, the non-linear H_∞ controller may exhibit a more dramatic onset of instability than the linear H_∞ controller [114].

The alternative implementation for the non-linear H_∞ control is to use MPC for controlling non-linear plants through their linearized models. The theoretical proof that a non-linear plant can be replaced by its linearized model to achieve local stability and robustness is given in [115], where it is shown that a non-linear system will be internally stable with a certain level of disturbance attenuation if its linearized form is stabilizable with the same level of disturbance attenuation. The linearization process of the general non-linear plant model can be summarized as follows. The general non-linear plant model introduced by the equations (5.9) and (5.10), is repeated here

$$x_{n+1} = f_n(x_n, u_n, \omega_n), \qquad x_0 = 0 \tag{7.2}$$

$$y_n = h_n(x_n, \omega_n) \tag{7.3}$$

with the assumption of $0 = f_0(0, 0, 0)$ and $0 = h_0(0, 0)$. This equation can be rewritten in the equivalent matrix function form

$$x_{n+1} = \tilde{A}(x_n) \cdot x_n + \tilde{B}(x_n) \cdot u_n + \tilde{D}(x_n) \cdot \omega_n, \qquad x_0 = 0 \tag{7.4}$$

$$y_n = \tilde{C}(x_n) \cdot x_n + \tilde{E}(x_n) \cdot \omega_n \tag{7.5}$$

where $\tilde{A}(x_n)$ $\tilde{B}(x_n)$ $\tilde{D}(x_n)$ $\tilde{C}(x_n)$ and $\tilde{E}(x_n)$ are continuous matrix functions. To obtain the standard control system formulation, the regulated output equation should be included

$$z_n = \tilde{H}(x_n) \cdot x_n + \tilde{G}(x_n) \cdot u_n . \tag{7.6}$$

The linearization process is performed by the linear estimation process in a way that the non-linear equations (7.4) and (7.5) are replaced by the linearized ones. In MPC, this linearization is a process performed by system identification

$$T_i : \quad \{\tilde{A}(x_n), \tilde{B}(x_n), \tilde{C}(x_n)\}_{n=0}^{N_C} \rightarrow \{A, B, C\} \tag{7.7}$$

where T_i represents the selected strategy for identification, described in "Parameter Estimation Strategies" page 53. The matrices D and E, as well as the H and G, are design parameters selected during the control system design. The plant model Σ, in discrete-time linear state space equation form is defined by the difference state equation and by the two output equations as introduced in (5.22) and repeated here

$$\Sigma : \quad \begin{aligned} x_{n+1} &= A \cdot x_n + B \cdot u_n + D \cdot \omega_n \\ y_n &= C \cdot x_n + E \cdot \omega_n \\ z_n &= H \cdot x_n + G \cdot u_n \end{aligned} \quad . \tag{7.8}$$

It is shown in [116] and [115] that a non-linear H_∞ control problem is locally solved by the linear H_∞ control law, synthesized in (7.8) by the linearization of (7.4) through (7.6) or equivalently (7.2) and (7.3).

7.3 Essentials of the Recursive Implementation

The purpose of this section is to define the implementation of an output feedback MPC, determined by the objective function (6.5). However, in the MPC environment, some assumptions that led to the formulation of (6.5), are no longer valid. For example, the term $-\gamma^2 \cdot x_0^T \cdot X^F \cdot x_0$ describes the assumption that at the initial time the state of the plant is not known and it is handled as a disturbance under the control of maximizing player. With MPC, the estimation of the state x_n is a continuous process and the estimation of the plant state $\{\ldots, x_{n-2}, x_{n-1}\}$ is available up to the present time $t = n \cdot T_s$, see Figure 7-1. To include this knowledge of the states, the objective function is modified in a way that the optimization does not start in the present, but rather in the past. The modified objective function (6.5) is [19]

$$J_\gamma(u, \omega) = x_N^T \cdot X^C \cdot x_N + \qquad . \qquad (7.9)$$

$$\sum_{k=n-N_0}^{N-1} \left\{ \begin{bmatrix} x_k^T & u_k^T \end{bmatrix} \cdot \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \cdot \begin{bmatrix} x_k \\ u_k \end{bmatrix} - \gamma^2 \cdot \omega_k^T \cdot \omega_k \right\} - \gamma^2 \cdot x_{n-N_0}^T \cdot X^F \cdot x_{n-N_0}$$

The objective function results in the backward and forward sequence of the Riccati equation solutions

$$J_\gamma(u, \omega) \rightarrow \begin{matrix} \{X^C = P_{n+N}^C, \ldots, P_{n+N_C}^C, \ldots, P_n^C, \ldots, P_{n-N_0}^C\} \\ \{X^F = P_{n-N_0}^F, \ldots, P_n^F, \ldots, P_{n+N_C}^F, \ldots, P_{n+N}^F\} \end{matrix} . \qquad (7.10)$$

As a consequence of MPC, the resulting dynamic feedback control is only used for the time period $[n \cdot T_s, (n+N_c) \cdot T_s]$. If the lengths of the sequences $N - N_C$ and N_0 are selected sufficiently large, it is expected that the recursive Riccati equation asymptotically converges to the solution of the Algebraic Riccati equations P^C and P^F, due to the fact that in every optimization period of the MPC, the model is not time varying

$$P^C_{n+k} = \begin{cases} X^C & k = N \\ \text{converges to } P^C & k = \{N-1, N_C\} \\ P^C & k = \{N_C, -N_0\} \end{cases} \tag{7.11}$$

$$P^F_{n+k} = \begin{cases} X^F & k = -N_0 \\ \text{converges to } P^F & k = \{-N_0+1, 0\} \\ P^F & k = \{0, N\} \end{cases}. \tag{7.12}$$

The sequences of the Riccati recursions $\{P^C_{n+N_C}, \ldots, P^C_n\}$ and $\{P^F_n, \ldots, P^F_{n+N_C}\}$, on the interval $[n \cdot T_s, (n+N_c) \cdot T_s]$, which are converging to the solution of the Algebraic Riccati equations P^C and P^F respectively, can be replaced by P^C and P^F in the dynamic control action calculation

$$\begin{aligned} \{P^C_{n+N_C}, \ldots, P^C_n\} &\Leftrightarrow \{P^C\} \\ \{P^F_n, \ldots, P^F_{n+N_C}\} &\Leftrightarrow \{P^F\} \end{aligned}. \tag{7.13}$$

Furthermore, there can only be a limited change in the plant model between two optimizations because these optimizations occur within short intervals. If this is taken into account, instead of initializing with the final X^C and initial X^F constrains, the Riccati recursion can be initialized with the results of the previous optimizations P^C and P^F.

Moreover, due to the fact that the optimization is performed on a constant plant model, the forward or backward optimization is meaningless and the backward optimization can be replaced with a forward recursion to calculate the algebraic Riccati solution. The Riccati recursion Ric( ) is initialized with the solution from the previous recursion $P^{C'}$ and $P^{F'}$ for the controller calculation

$$P^C = P_k^C = \mathrm{Ric}(P_{k-1}^C) \qquad \text{until} \qquad P_k^C - P_{k-1}^C \approx 0 \qquad P_0^C = P^{C'} \tag{7.14}$$

and for the state observer calculation

$$P^F = P_k^F = \mathrm{Ric}(P_{k-1}^F) \qquad \text{until} \qquad P_k^F - P_{k-1}^F \approx 0 \qquad P_0^F = P^{F'}. \tag{7.15}$$

In every control period, the new dynamic control parameters have to be re-initialized, similarly as presented in [117]. At the start of the control period the collected observations from the plant input $\{u_{n-N_0}, \ldots, u_{n-1}\}$ and the corresponding response $\{y_{n-N_0}, \ldots, y_{n-1}\}$ are used for the re-initialization. The estimator for the dynamic feed-back defined in (6.2) is re-tuned by the observations

$$\begin{gathered} k = -N_0, \ldots, 0 \\ \hat{x}_{n+k+1} = F^x \cdot \hat{x}_{n+k} + F^u \cdot u_{n+k} + F^y \cdot y_{n+k} \qquad \hat{x}_{n-N_C} = 0 \end{gathered}. \tag{7.16}$$

It is required that the selected N_0 is large enough for the recursion in the equation (7.16) to converge to the state of the plant x_n at the start of the present control sequence.

Part - III Simulation Studies

Part I of this thesis has defined the theoretical background for system identification, while in Part II an overview of the control system implementation is provided. In Part III the validity of the theory is examined through simulation studies.

Simulation techniques are very useful in the planning and design of modern power systems. They allow researchers to conduct inexpensive and reliable experiments involving various variables, scenarios, and techniques. The advantage of simulation studies lies in their ability to invoke unexpected and unforeseen circumstances and to simulate situations before they happen in the real physical plant. Development of new numerical computation techniques and fast computers has made it possible to simulate large power systems comprised of thousands of generators.

In this Part, the objective is to test the correctness and effectiveness of the proposed control algorithm as a power system stabilizer. This is considered as a challenging task due to the non-linear nature of the power system. At this stage, the testing is conducted on the simulated plant model, before it is performed on a physical system.

8 Single-Machine Infinite-Bus System

Electrical energy has become a major form of energy for end use consumption in today's society. To make electric power generation and transmission more economic and reliable, the power station sites are selected close to a source of energy, such as large coal mines and water power sources. This has resulted in an interconnected network of transmission lines, linking generators and loads into a large integrated system. In such a system, the power transmission lines and networks have to meet the following requirements:

1. ensure parallel operation of the thousands of distant power stations connected together
2. transmit electrical power to large load centers over large distances
3. ensure satisfactory parallel operation with other interconnected power systems.

For proper operation, this large integrated system requires a stable operating condition. Stability in power systems is generally regarded as the ability of generating units to maintain synchronous operation [26].

A power system is a sophisticated combination of multiple electrical and mechanical components. In general, these elements are highly non-linear. Also, power system operation is characterized by a wide range of operating conditions, random load changes and various unpredictable disturbances, which make it an ideal test bench for the proposed supervised system parameter identification [118-119]. The power system simulation package used in this research was developed at the University of Calgary in the early 1990s, and used in numerous simulation studies. A description of this package can be found in [120].

To understand the dynamic behavior of an electric power system and to design a controller to improve its stability, it is necessary to first become familiar with the basic components of the system and to understand the simplest connection between its elements. The single-machine infinite-bus system, as shown in Figure 8-1, is the simplest form of a power system [121].

8.1 Introduction

The power system stabilizer (PSS) is used to modulate the generator's excitation in such a way as to provide additional damping to the electromechanical oscillations of the generating unit, thereby improving the steady-state stability of the power system. To meet this expectation, the PSS has to produce a component of electrical torque in the synchronous machine that is in phase with the speed variation of rotors. Studies have shown that the use of supplementary control signal in the excitation system and/or governor system of the generating unit can provide extra damping for the system and thus, improve the unit's dynamic performance [122].

The PSS, first proposed in the 1950's, is based on a simplified model of the power system, linearized at some operating point, to damp the low frequency oscillations of the power system. The most widely used PSS, known as the Conventional PSS (CPSS), is a fixed parameter lead-lag type control device that has made a great contribution in enhancing power system dynamic stability [123]. See Appendix A.3.

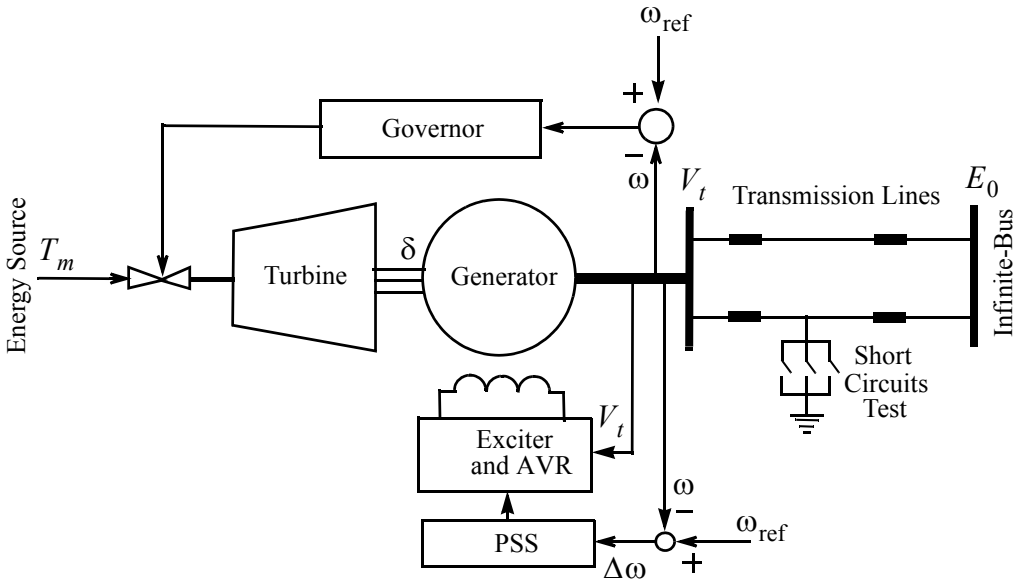


Figure 8-1 System model used in the simulation studies

The complexity of power systems is increasing continuously. At the same time securing uninterrupted electricity supply for the customers is increasing in importance. This increasing demand for quality electricity in a fundamentally unstable environment led to research in using modern control techniques to improve stability. Since the power system configuration keeps changing, either due to switching actions in the short term or system enhancements in the long term, it would be more suitable to use adaptive control techniques that can track the operating conditions and changes in the system. An Adaptive PSS (APSS) can adjust its parameters on-line according to the environment in which it operates and can provide good damping over a wide range of operating conditions [124-125]. In the following, the implementation of model predictive control (MPC), introduced in Parts I and II, as a PSS is examined.

8.2 Multimodel System Identification

A simulation study was conducted using the H_∞ MPC as an adaptive power system stabilizer. The tests were performed on a UNIX based power system model. The results are presented next.

8.2.1 System Model

In the proposed H_∞ MPC, the system model is identified by a fifth order discrete time model of the form

$$A(q^{-1}) \cdot y_n = B(q^{-1}) \cdot u_n + v_n \tag{8.1}$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the backward shift operator q^{-1} and are defined as

$$A(q^{-1}) = 1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + a_3 \cdot q^{-3} + a_4 \cdot z^{-4} + a_5 \cdot q^{-5} \tag{8.2}$$

$$B(q^{-1}) = b_1 \cdot q^{-1} + b_2 \cdot q^{-2} + b_3 \cdot q^{-3} + b_4 \cdot z^{-4} + b_5 \cdot q^{-5} \tag{8.3}$$

and variables y_n, u_n and v_n are the system output, system input and the model prediction error, respectively.

The control response is computed based on the identified model parameters. The correctness of the identification method determines the preciseness of the identified model, which tries to reflect the true system. There are several methods which can be used to

obtain an estimate for the model parameters. For example, the RLS, LMS and Kalman Filtering methods are well known and in widespread use.

To enhance the system identification and to omit the difficult task of choosing one particular parameter estimation method, the idea introduced in Part I is implemented. Instead of one algorithm to estimate the system parameters, two or more algorithms should be used, which compete with each other, and an expert system does the selection between them.

The implemented algorithms. In this study eight algorithms, developed according to the definitions in Part I, are implemented:

1. **PRO** - the Proposed model. In this algorithm the parameters are kept fixed which are the estimated parameters, selected at the last decision. These parameters are used by the H_∞ controller to calculate the control action.
2. **RLS** - the well known Recursive Least Squares algorithm from Section 3.1.
3. **LMS** - the Least Mean Squares algorithm from Section 3.2.
4. **WCE** - the Worst-Case Estimation algorithm, based on the linear programming method, from Section 3.3.
5. **KF** - the famous Kalman Filter based system identification from Section 3.4.
6. **FQR** - a mathematically more stable Recursive Fast QR RLS algorithm from Section 3.5.
7. **IQR** - another version of a mathematically more stable Inverse QR RLS algorithm from Section 3.5

8. **KFA -** a modified version of the Kalman Filter algorithm from Section 3.4.

When the power system is under transient conditions, such as configuration changes or operating condition changes, the identifier is required to track such changes quickly. This can be achieved with a short decision period, $T_d = N \cdot T_s$, where for $N = 183$ and for a $T_s = 0.025s$ sampling interval the time between two decisions is $T_d = 4.575s$. This is illustrated in Figure 2-7 and Figure 7-2. An odd number for $N = 183$ is randomly selected to avoid coincidences with the disturbances during simulations.

The expert system for the selection process calculates the moving average, the quadratic moving average, and the difference between the maximum and minimum error of the residuals. In the following this process is presented for an array of perturbations.

Generator's operating point. In order to investigate the system identification stability and reliability, the generator of the single-machine infinite-bus system is operated under a full load condition $P = 0.97pu$ and $pf = 0.97lag$. A sequence of different perturbations is applied to the infinite-bus system as presented in Table 8-1. This sequence of disturbances is selected to simulate a worst-case situation.

Table 8-1 Disturbance sequence for the system identification tests

T_occur [s]	Intensity	Disturbance type
256	0.06 pu	Mechanical power reference step change
276	-0.03 pu	Mechanical power reference step change
285	-0.03 pu	Mechanical power reference step change
381	0.1 s	Bus three phase to ground fault
411	-0.1 s	Exciter voltage reference step change

Table 8-1 Disturbance sequence for the system identification tests

T_occur [s]	Intensity	Disturbance type
435	0.07 pu	Exciter voltage reference step change
450	0.03 pu	Exciter voltage reference step change
561	-0.18 pu	Exciter voltage reference step change
571	0.09 pu	Exciter voltage reference step change
587	0.09 pu	Exciter voltage reference step change
621	0.1 s	Exciter voltage reference step change
681	0.1 s	Exciter voltage reference step change
711	0.05 pu	Mechanical power reference step change
720	-0.05 pu	Mechanical power reference step change
801	0.1 s	Bus three phase to ground fault
851	0.1 s	Bus three phase to ground fault
935	-0.05 pu	Exciter voltage reference step change
975	0.05 pu	Exciter voltage reference step change

8.2.2 Moving Average Calculation

The moving average, v_{moa}^{k}, is calculated for every decision period, $T_d = N \cdot T_s$, as a sum of residuals for all eight algorithms

$$v_{\text{moa}}^{k} = \frac{1}{N} \cdot \sum_{i=n-N}^{n} v_i^k \qquad \text{where} \qquad v_i^k = \phi_i^T \theta_i^k - y_i \qquad k = 1, 2\ldots, 8 \quad . \quad (8.4)$$

Simulation results are presented in Figure 8-2 as absolute values of the residuals from the eight algorithms. In this figure, the consistently good performance of the PRO algorithm is shown by the dark line. It is also interesting to point out that the KFA model behavior, presented by the dotted line of the highest magnitude, is not stable. This instability is detected by the expert system and it is discounted from the PRO model.

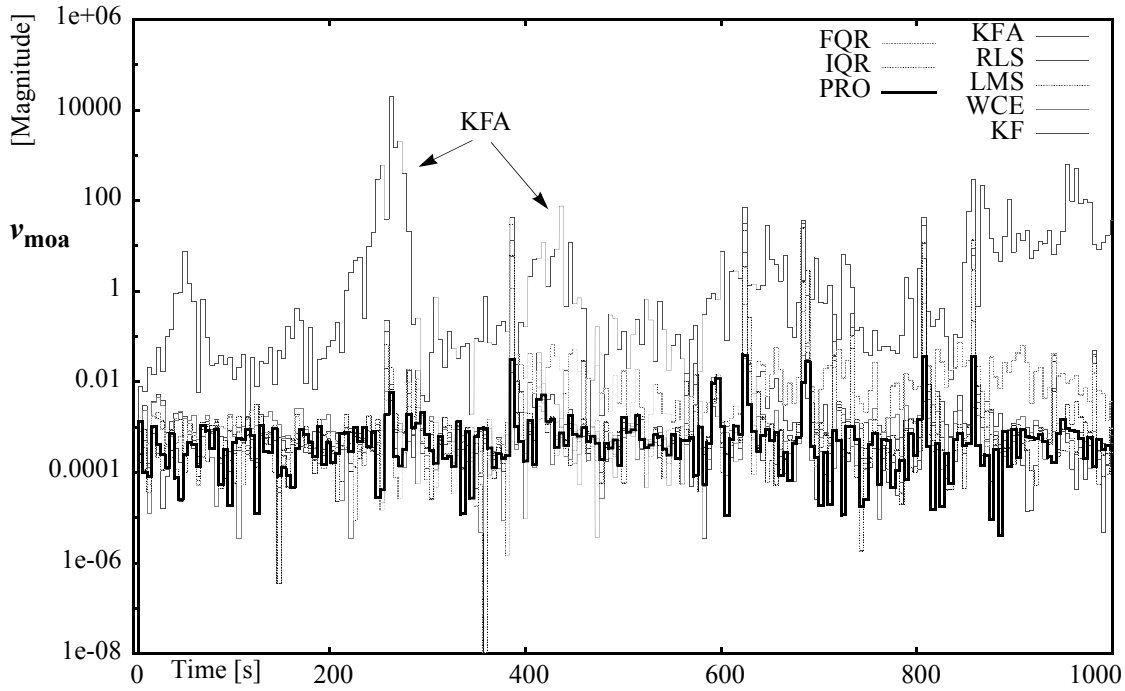


Figure 8-2 Moving average of the residuals

8.2.3 Quadratic Moving Average Calculation

The quadratic moving average, v_{qma}^{k}, is calculated for every decision period, $T_d = N \cdot T_s$, as a sum of square of residuals for all eight algorithms

$$v_{\text{qma}}^{k} = \frac{1}{N} \cdot \sum_{i=n}^{n+N} (v_n^k)^2 \qquad \text{where} \qquad v_n^k = \phi_n^T \theta_n^k - y_n \qquad k = 1, 2\ldots, 8. \qquad (8.5)$$

The results from the simulation, in Figure 8-3, clearly show that the residuals of PRO algorithm have very low values of quadratic moving average. This further indicates that the selection of the expert system results in a proper system model.

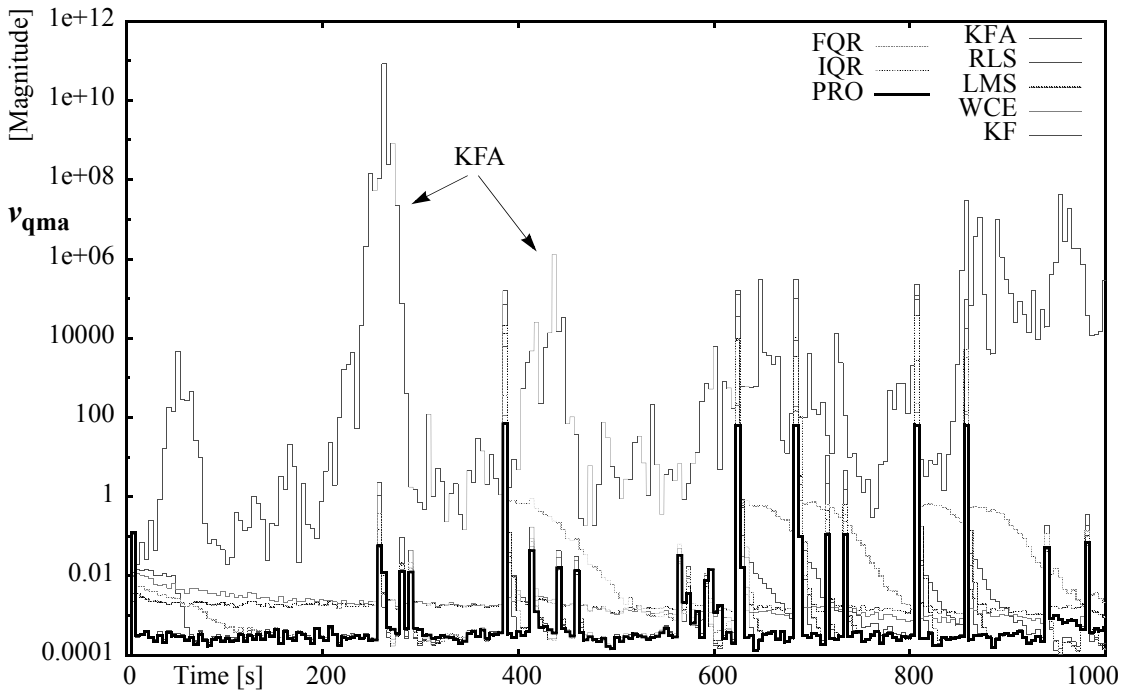


Figure 8-3 Quadratic moving residuals

8.2.4 Maximum Residual Amplitude

The maximum residual amplitude, v_{mer}^{k}, is calculated for every decision period, $T_d = N \cdot T_s$, as a difference between the maximum and minimum (negative) residuals for all eight algorithms

$$v_{\text{mer}}^{k} = \max_{i \in \{n, n+N\}} \left(v_i^k \right) - \min_{i \in \{n, n+N\}} \left(v_i^k \right) \qquad k = 1, 2 \ldots, 8 . \tag{8.6}$$

The maximum residual is the largest prediction error which occurs in the decision interval, and the minimum residual is the smallest (negative) prediction error. The differ-

ence of these two values will always be a positive number, which is in some sense proportional to the H_∞ norm of the system identification. From Figure 8-4 it can be observed that the selected PRO algorithm's maximum residual amplitude is among the lowest one during the simulation.

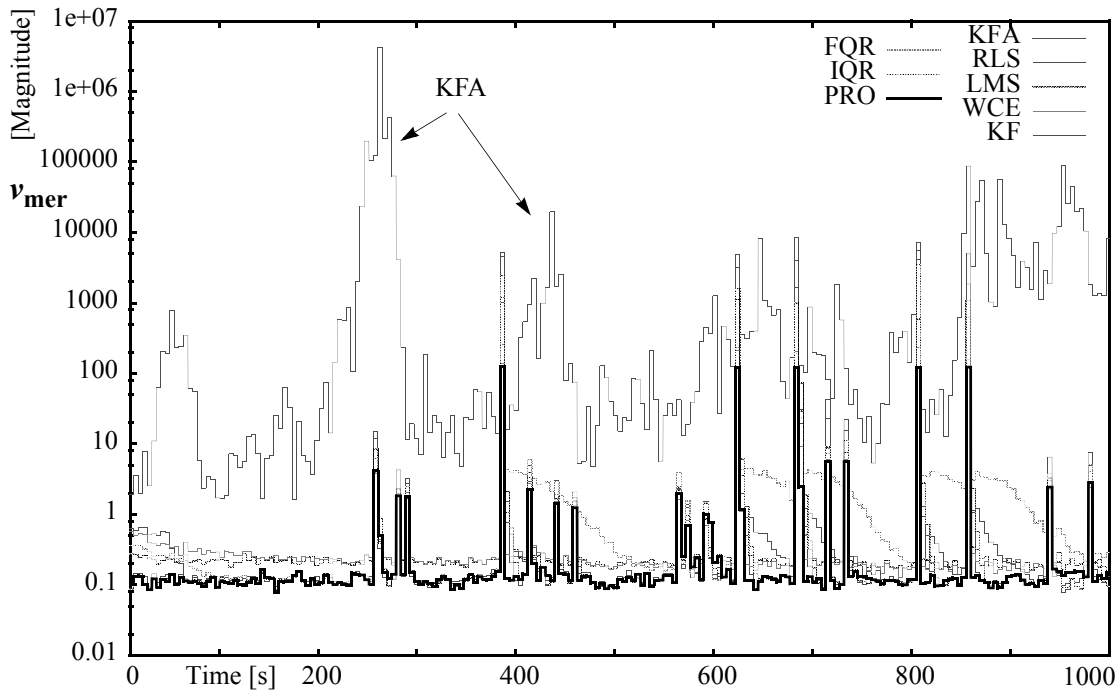


Figure 8-4 Maximum residual amplitude

8.2.5 Parameter Vector's Mean Square Error Estimation

Estimating the mean square error, (*mse*) v_{mse}^k, of the estimated parameter vector, θ_n^k, in accordance with the theory introduced in section 2.6.2, is described below

$$\theta_{\text{moa}}^{k} = \sum_{i=n}^{n+N} \theta_{i}^{k} \qquad \text{for} \qquad k = 1, 2\ldots, 8 \tag{8.7}$$

$$\theta_{\text{qma}}^{k} = \sum_{i=n}^{n+N} (\theta_{i}^{k})^{2} \qquad \text{for} \qquad k = 1, 2\ldots, 8 \tag{8.8}$$

$$\theta_{\text{var}}^{k} = \frac{1}{N} \cdot (\theta_{\text{qma}}^{k} - (\theta_{\text{moa}}^{k})^{2}) \qquad \text{for} \qquad k = 1, 2\ldots, 8 \quad . \tag{8.9}$$

The obtained parameter variances, θ_{var}^{k}, are matrix (vector) values which cannot be easily compared to each other in the decision process. To overcome this difficulty, a norm of the variance vector is calculated. One of the most frequently used matrix norm in numerical linear algebra is the Frobenius[1] norm [57]

$$\left\| \theta_{\text{var}}^{k} \right\|_{F} = \sqrt{\sum_{i=1}^{\text{row}} \sum_{j=1}^{\text{col}} \theta_{\text{var}}^{k}(i,j)} \qquad \text{for} \qquad k = 1, 2\ldots, 8\,. \tag{8.10}$$

This norm is implemented to evaluate the quality of the identified system parameters. For comparison, this norm is presented in Figure 8-5. The curve with the highest magnitude belongs to the AKF algorithm because during this simulation, the AKF algorithm is not a reliable identification method.

The two curves below the PRO line belong to the WCE and LMS algorithms and suggest slower adaptation and less variance in the identified parameters.

1. George Frobenius, 1849-1917, Berlin, Germany, Mathematician

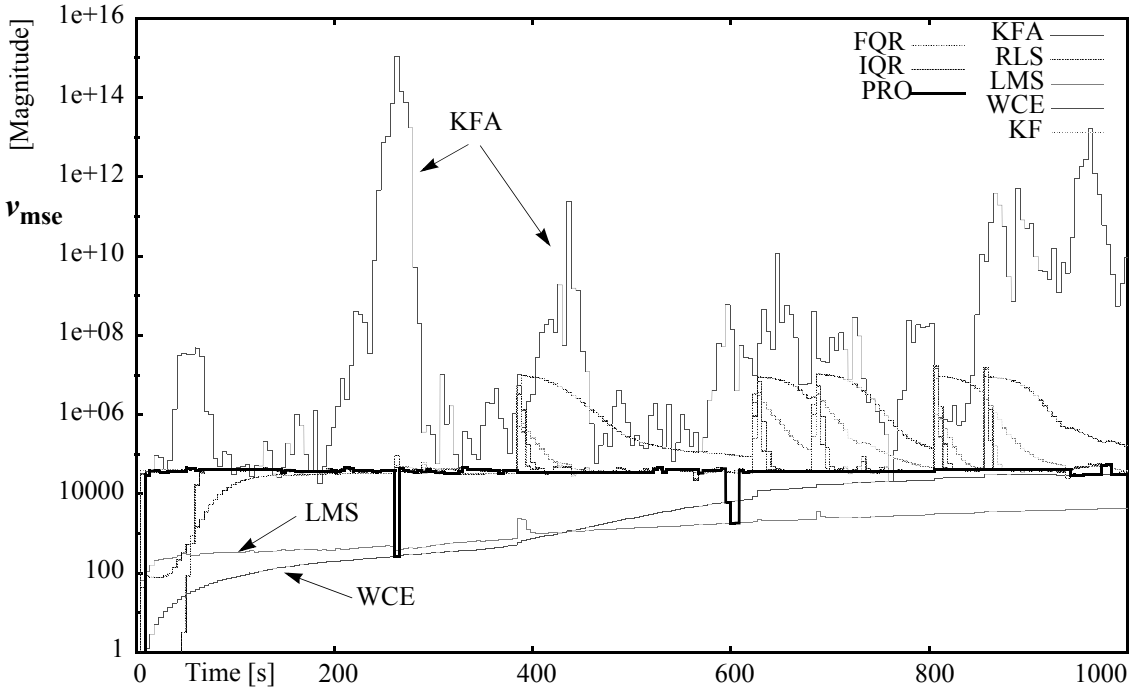


Figure 8-5 Forbenius norm of the identified parameter variance

The analysis of the simulation results shown in Figure 8-2 to Figure 8-5 led to the conclusion that the expert system decision should be satisfactory if only v_{qma}^{k} and v_{mer}^{k} values are used in the decision process. In addition, it is not necessary to implement the mean square error, (*mse*) v_{mse}^{k}, of the estimated parameter vector, θ_{n}^{k}, in the decision process because the analysis of the residuals well represents the quality of the identified model. This simplification significantly reduces the necessary calculations.

8.2.6 Reminiscence Function

The suggested reminiscence function, introduced in Section 2.6.3, can be effectively implemented using the exponentially weighted moving average (EWMA) function, which is calculated for every decision period, $T_d = N \cdot T_s$,

$$J_n^k = \lambda \cdot J_{n-N}^k + (1-\lambda) \cdot v_{\text{qma}}^k \cdot v_{\text{mer}}^k \qquad \text{for} \qquad k = 1, 2\ldots, 8 \qquad (8.11)$$

where J_{n-N}^k is the last calculated performance index at time $t = (n-N) \cdot T_s$, λ is the reminiscence factor, v_{qma}^k and v_{mer}^k are the calculated values from (8.5) and (8.6) respectively. The reminiscence factor shall be selected from the interval $0 \le \lambda < 1$. If $\lambda = 0$ the decision will be based on the latest residual analysis only and the decision will ignore any past experience. If $\lambda \to 1$ the decision is more reliant on the past behavior of the residuals. In this research $\lambda = 0.3333$ was selected as a compromise solution, since it results in a less frequent switching between the parameters. However, it allows the selection of new parameters when the system changes.

In Figure 8-6 the comparison of the performance index (8.11) is presented. As expected, the selected model represented by the PRO algorithm has one of the best performance indices.

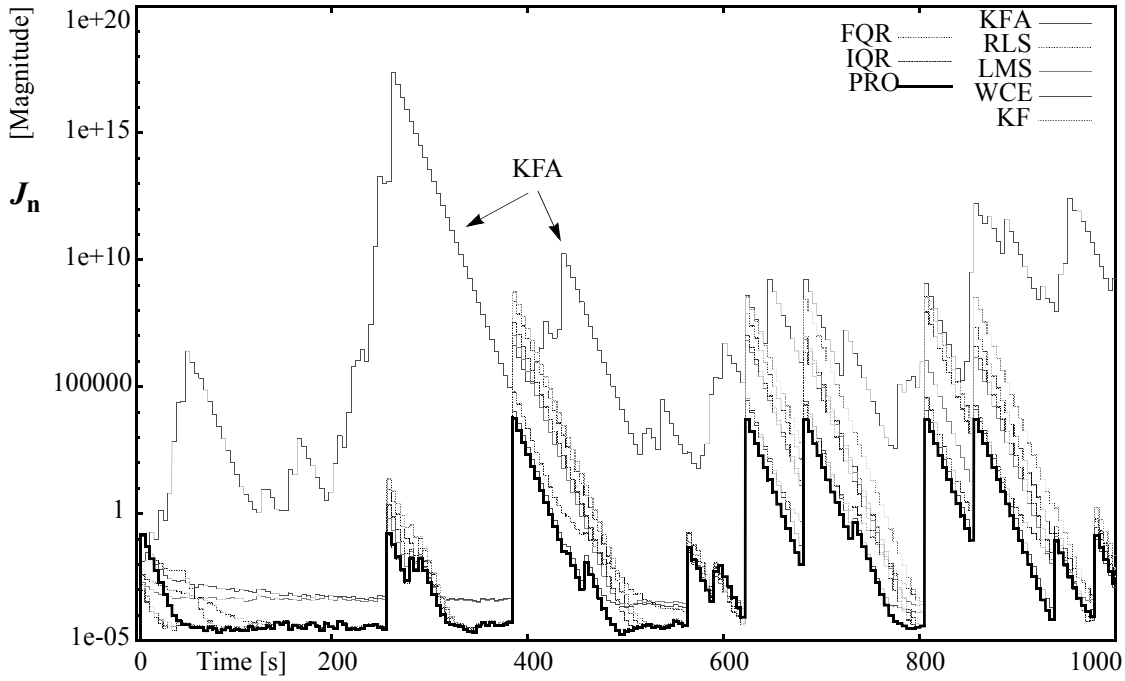


Figure 8-6 Performance index for all algorithms

The expert system based on the calculated performance index selects the system parameters from one of the eight identification algorithms, for the next $T_d = N \cdot T_s$ period. The selected parameters are loaded into the PRO model and are used by the control algorithm in that same period. To illustrate this selection process Figure 8-7 is presented, where the "y" coordinate indicates from which algorithm the parameters are selected. For example, at the start, t=0, the PRO algorithm is selected, which at that time contains zero elements. The first selected identified parameters belong to the 7^{th} algorithm (IQR), which is selected few times in a row. After that, the expert system selects the same parameters for a while, which are identified by IQR, and so on. It is interesting to mention that in this particular simulation the Kalman Filter (KF) algorithm is never selected. However, in some

other simulations, it is favored. Finally, due to unreliable operation, the AKF algorithm is never selected.

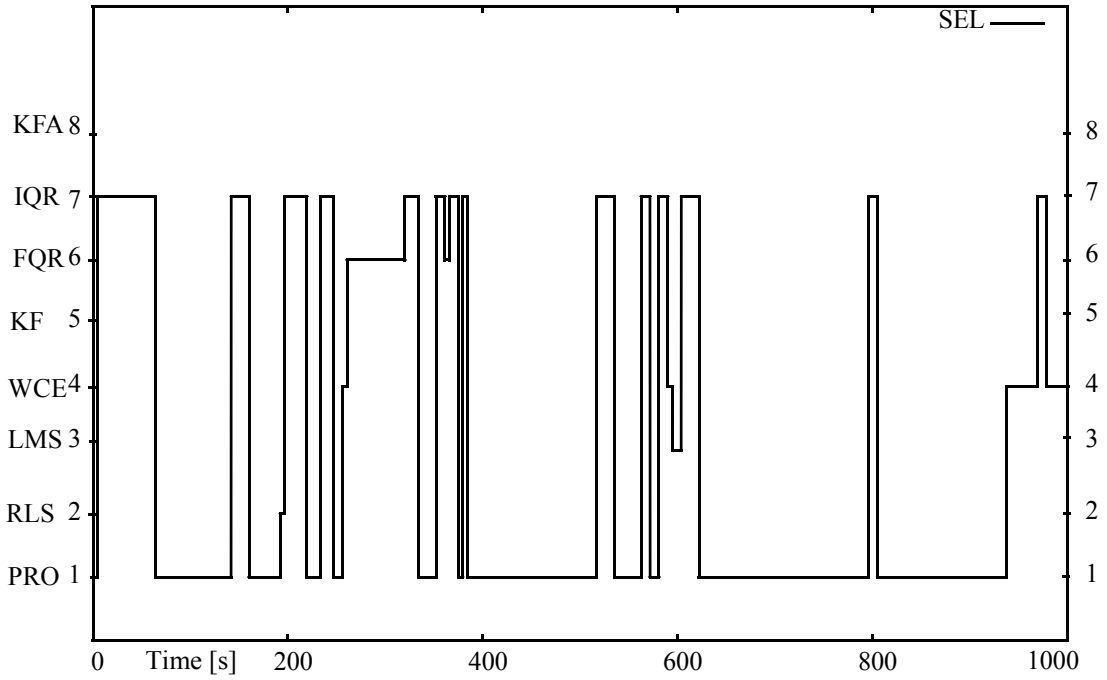


Figure 8-7 System model selection index

Figure 8-8 and Figure 8-9 show the typical curves of the identified parameters as a function of time for the 5th order linear system model

$$\hat{y}_n = [-y_{n-1}, \ldots -y_{n-5}, u_{n-1}, \ldots u_{n-5}] \cdot \begin{bmatrix} a_1 \\ \ldots \\ a_5 \\ b_1 \\ \ldots \\ b_5 \end{bmatrix}. \tag{8.12}$$

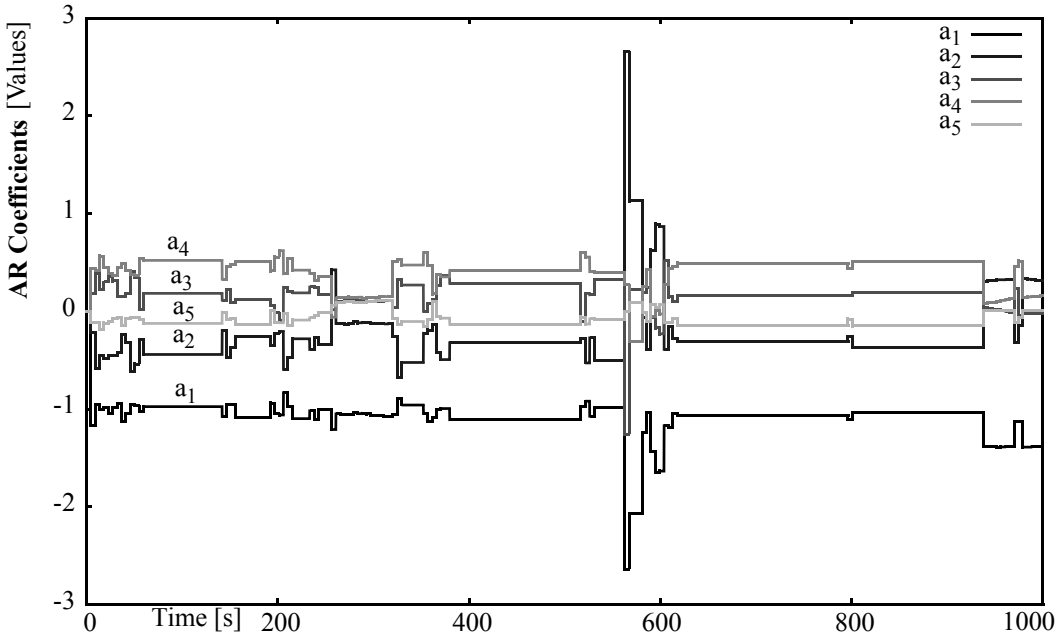


Figure 8-8 Identified a_i parameters

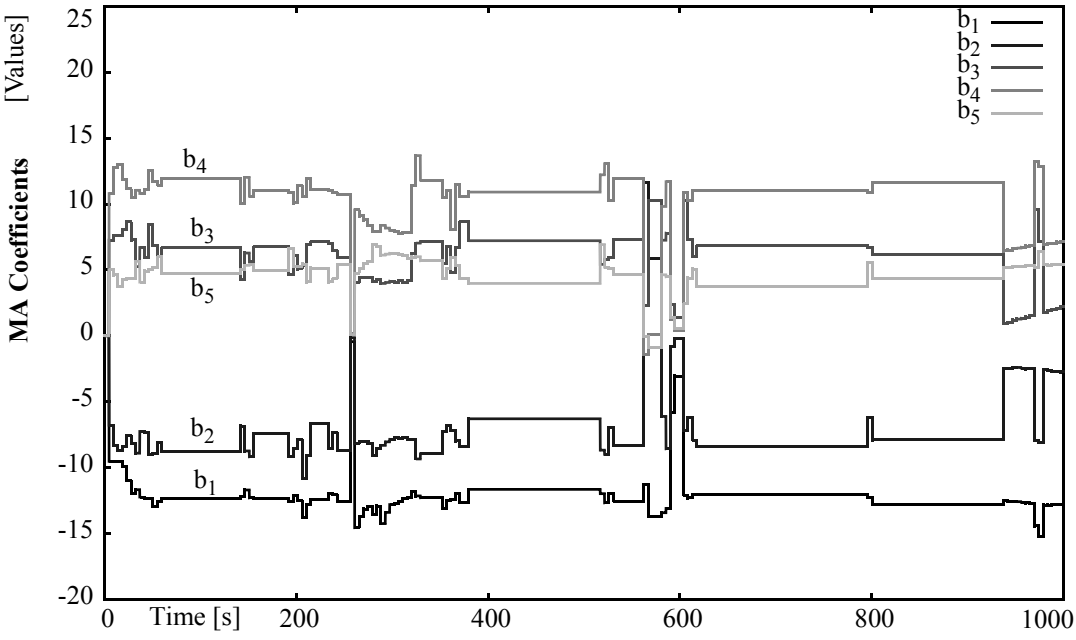


Figure 8-9 Identified b_i parameters

The squared residuals for the identified and selected model, presented in Figure 8-10, demonstrate steady quality. The spikes above the 0.01 level stem from the inability of the model to predict the external perturbations that are introduced during the simulation to test the MPC stability. It is evident that the model is keeping its stability under these perturbations.

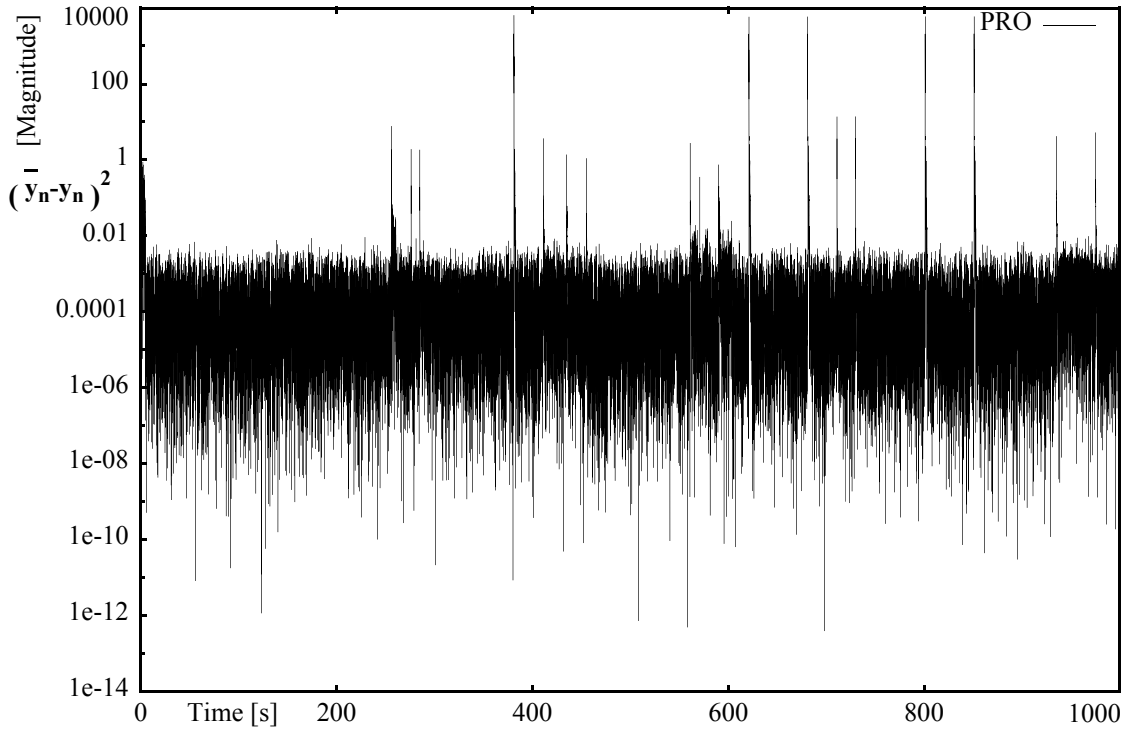


Figure 8-10 System model prediction error for the PRO model

8.3 Control Test Results

A number of tests have been performed to investigate the effect of the proposed H_∞ Model Predictive Controller (H_∞ MPC) on the power system stability. The results are compared with those of the Conventional Power System Stabilizer (CPSS) and No Controller (NoC). The CPSS is a two stage lead-lag type controller that is part of the simulation package, see Appendix A.3. The MPC is an H_∞ algorithm based digital controller with the sampling time of $T_S = 0.025s$. A block diagram of the study system is shown in Figure 8-1.

8.3.1 Weight Selection for H_∞ MPC

As explained in section 7.2, the H_∞ MPC adaptively selects its own system model for the control calculation. The control performance, however, is also determined by the selection of weight matrices. In the case of linear H_∞ (sub)optimal controller, the system performance depends on the weighting matrices D, E, H and G from (6.1). In the design, during the selection of weight matrices, a trade-off must be made to achieve stability and satisfactory control action.

It was found in the simulation studies that the control system of less than 5^{th} order does not provide adequate stabilizing response. When tested, the control system of 9^{th} order offered an excellent performance. However, as a compromise the minimum control sys-

tem dimension of the 5th order was selected, due to its computational requirements and performance.

Trade-off studies for the weight selection were conducted by repeatedly varying the weight parameters until the desired results were acquired. The closed-loop system obtained in this manner is then analyzed with respect to transient response, typical disturbance response, robustness, etc., to achieve the best system behavior.

For simplification, the weight matrices are selected to be diagonal, with the following final values

$$D = 0.06 \cdot I^{5\times 5}, \quad E = 0.0004 \cdot 1^{5\times 1}, \quad H = 0.0215 \cdot \begin{bmatrix} I^{5\times 5} & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad G = 1.0$$

where $I^{n\times n}$ is an $n \times n$ unity matrix, and $1^{n\times 1}$ is an n length vector of *1s*.

8.3.2 CPSS Parameter Tuning

In order to compare the dynamic performance of the power system stabilization with H_∞ MPC and CPSS, the CPSS parameters are tuned to an "optimal" value under normal system operating conditions. In the CPSS tuning test, when the generator is operating with $P = 0.97pu$ and $pf = 0.97lag$ a disturbance of 0.05 pu step increase and decrease of input torque reference is applied at time 1 s. Under these conditions, the CPSS parameters are tuned according to the tuning procedure described in [127-128]. The parameters of the CPSS are then kept unchanged for all the tests. Time response of the rotor speed deviation

with H_∞ MPC, CPSS and an open system response is shown in Figure 8-11. The selected CPSS parameters are presented in Appendix A.3.

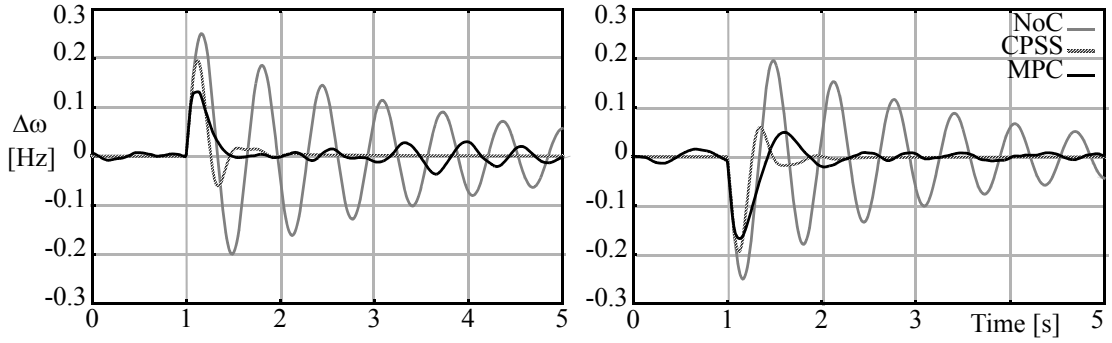


Figure 8-11 Mechanical power reference step change of +/- 0.05 pu

8.3.3 Full Load Test

The system state is observed by the controller through the generator's rotational speed deviation, $\Delta\omega$, as illustrated in Figure 8-1. To demonstrate the control action three characteristic examples are shown in Figures 8-13 through 8-17. In these tests, the generator of the single-machine infinite-bus system is operating under a full load condition $P = 0.97pu$ and $pf = 0.97lag$.

In the first test, a $\Delta P = 0.15pu$ step increase and decrease in input mechanical torque is applied at 1 s and 5 s respectively. These changes in the system can be best observed through the generator's electrical power, presented in Figure 8-12.

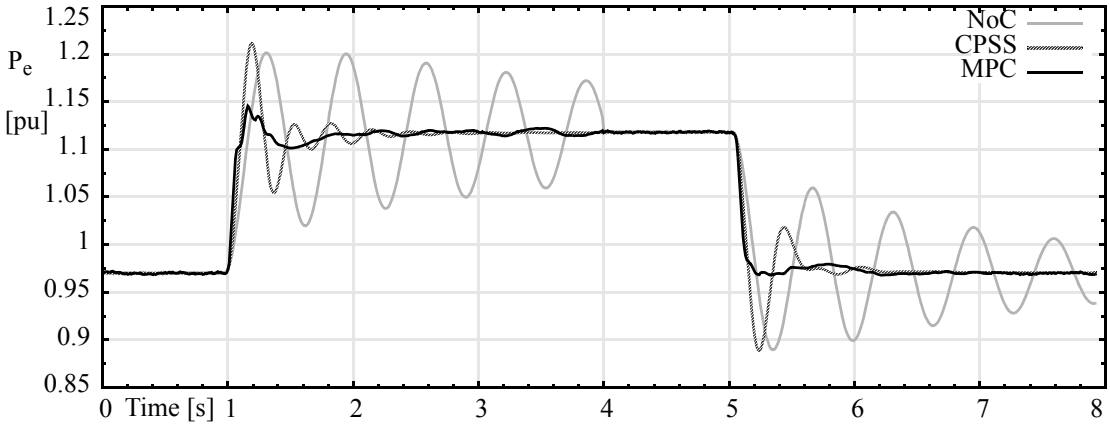


Figure 8-12 Electrical power change for mechanical torque change

This test is intended to demonstrate the control action of the H_∞ MPC for an "average" disturbance. The measured rotational speed deviation, $\Delta\omega$, of the generator with the controller's response is shown in Figure 8-13. It can be observed from this figure that the

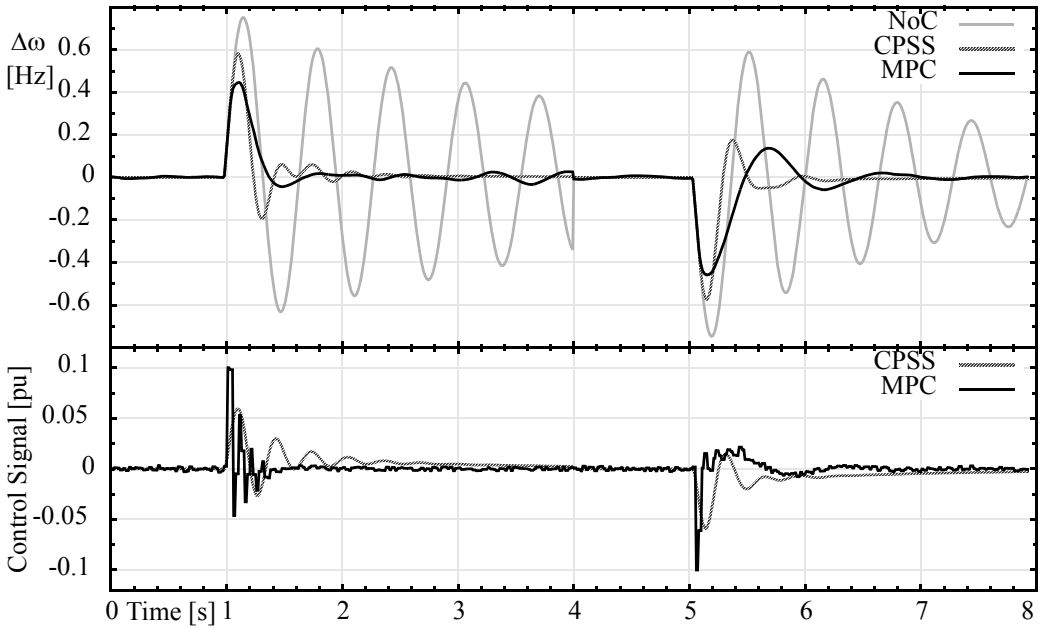


Figure 8-13 Control action for an average disturbance

control action of the H_∞ MPC is faster than that of the CPSS, which results in a lower peak of $\Delta\omega$.

To demonstrate the control action for a large disturbance, a three phase to ground fault test of 100 ms length on generator bus has been performed. This test is best characterized by the change in generator's terminal voltage, V_t, as presented in Figure 8-14.

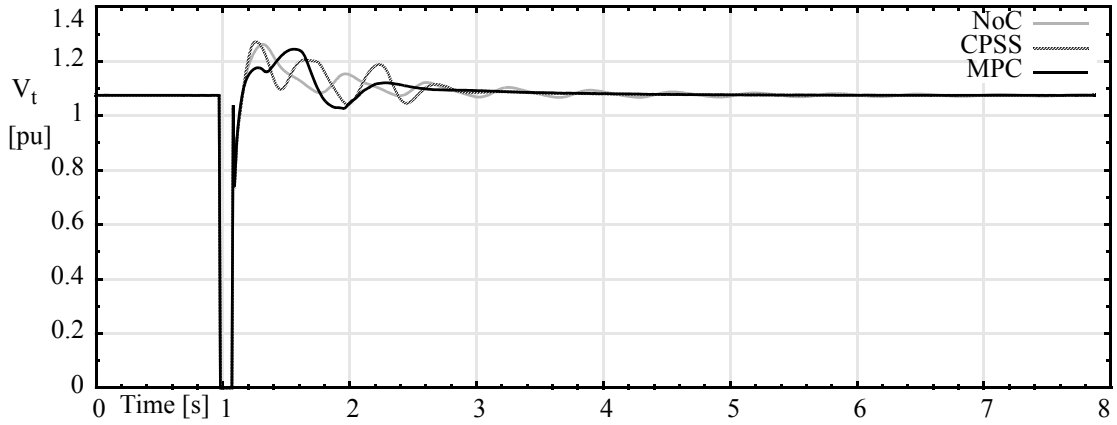


Figure 8-14 Terminal voltage during three phase to ground fault

The controller's response to the three phase to ground fault is shown in Figure 8-15. The control action of the H_∞ MPC is fast and strong. However, it is constrained by the 0.1 pu control signal limits. As expected from an H_∞ control, the H_∞ MPC significantly reduces the peak of the generator's rotational speed deviation after the fault, and it helps the system return to its operating point quickly.

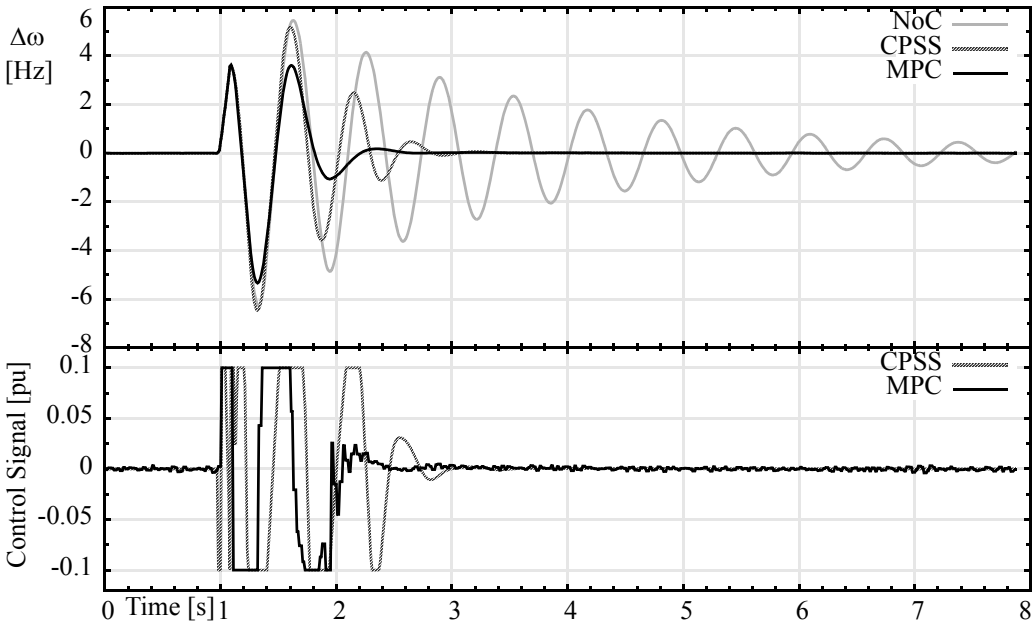


Figure 8-15 Control action for a large disturbance

To demonstrate the control action for small disturbances, a sequence of exciter voltage reference step change tests have been performed. Figure 8-16 shows the changes in the terminal voltage of the generator, V_t, when the exciter voltage reference is changed by

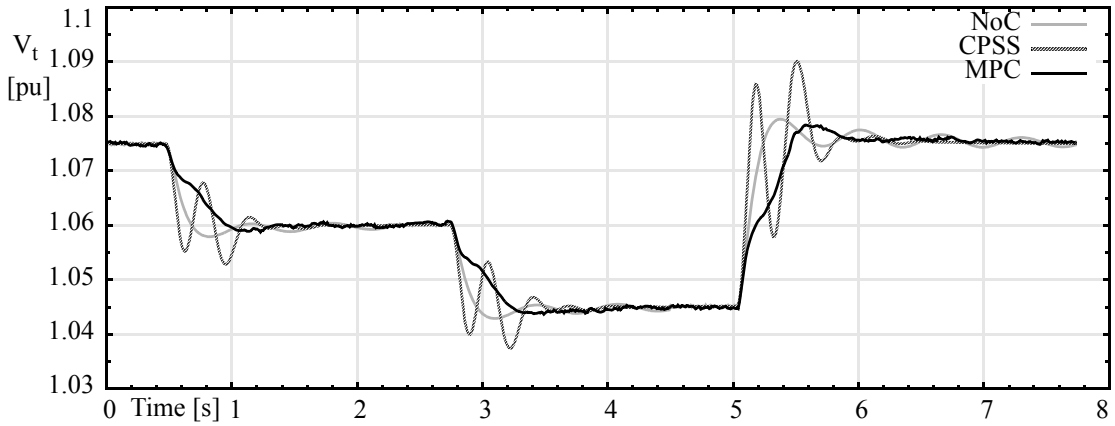


Figure 8-16 Terminal voltage for exciter reference changes

–0.015 pu, –0.015 pu and 0.03 pu steps, at 0.5 s, 2.8 s and 5 s respectively.

Figure 8-17 shows the generator's rotational speed deviation and control action for the exciter voltage reference step change. This test further demonstrates the effectiveness of

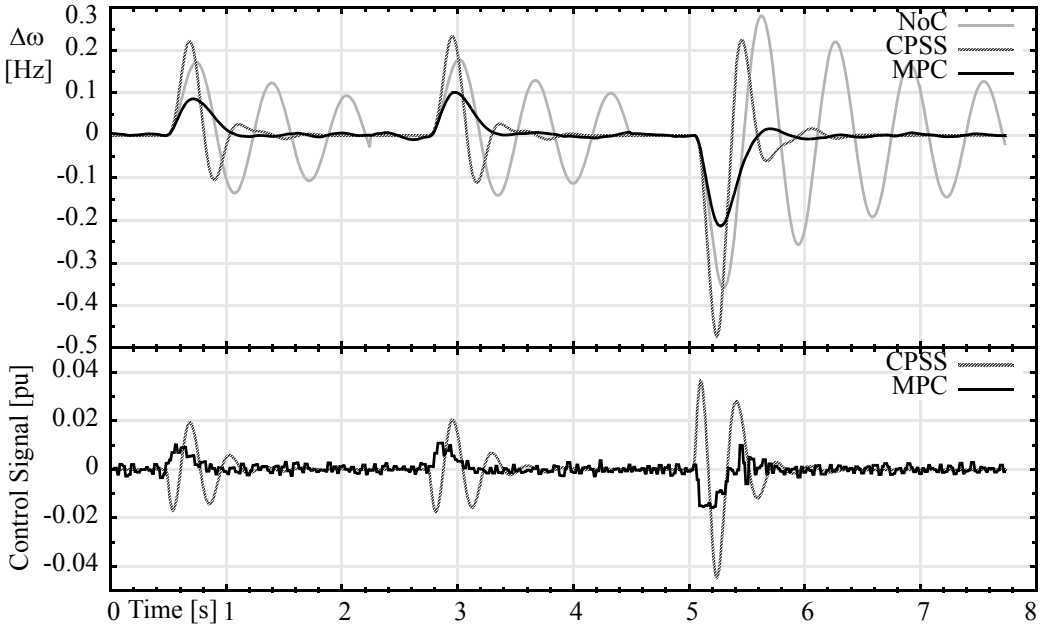


Figure 8-17 Control action for small disturbances

the H_∞ MPC controller, which significantly reduces the peak of the generator's rotational speed deviation. Contrary to this, the initial response of the CPSS to these smaller disturbances is confused.

8.3.4 Light Load Test

With the generator in a single-machine infinite-bus power system operating under light load condition, $P = 0.5pu$ and $pf = 0.87lag$, a 0.15 pu step increase and decrease in input mechanical torque is applied at 1 s and 5 s. The generator's rotational speed deviation, $\Delta\omega$, for this test is shown in Figure 8-18. The test results verify that both control

algorithms perform very well. The difference in performance goes in favor of the H_∞ MPC with smaller amplitudes in post-disturbance oscillations. However, the over-conservative behavior of the H_∞ controller can be observed in Figure 8-18, where the effects of disturbances with highest magnitudes are reduced, while the small oscillations are left uncontrolled.

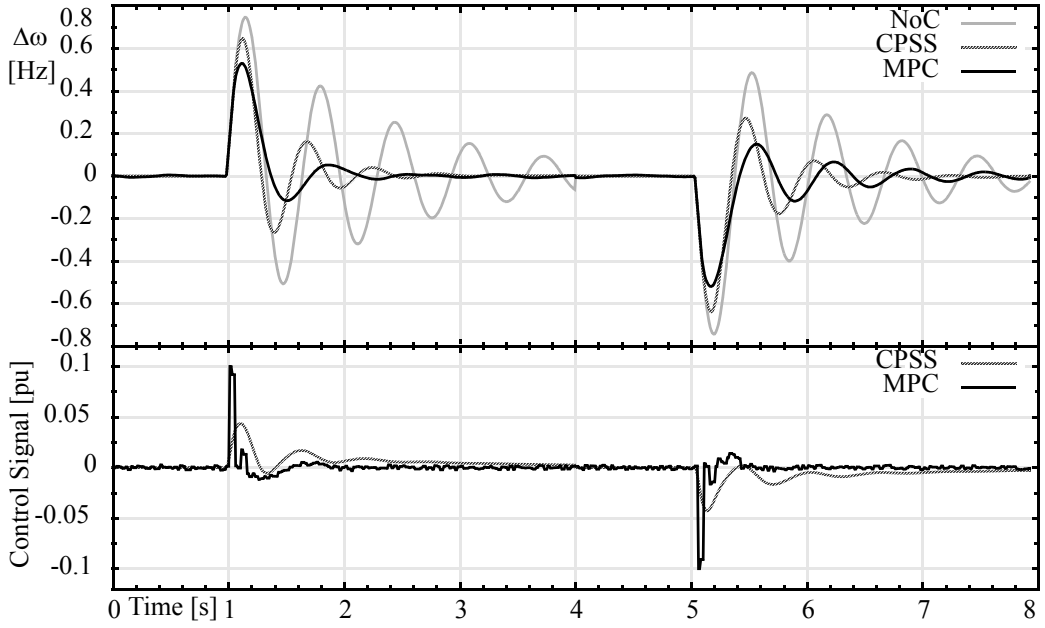


Figure 8-18 Response to a +/- 0.15 pu step change of torque in light load test

Further, under the same load conditions, $P = 0.5pu$ and $pf = 0.87lag$, a sequence of exciter voltage reference step changes of -0.015 pu, -0.015 pu and 0.03 pu is applied at 0.5 s, 2.8 s and 5 s respectively. The measured rotational speed deviation, $\Delta\omega$, of the generator with the controller's response is shown in Figure 8-19.

The CPSS parameters are selected as such to have the best performance under the load conditions when the generating unit is operated with $P = 0.97pu$ and $pf = 0.97lag$.

Under light load conditions, the stabilization effect of the CPSS cannot match the performance of H_∞ MPC which performs an adaptation for the changed load conditions.

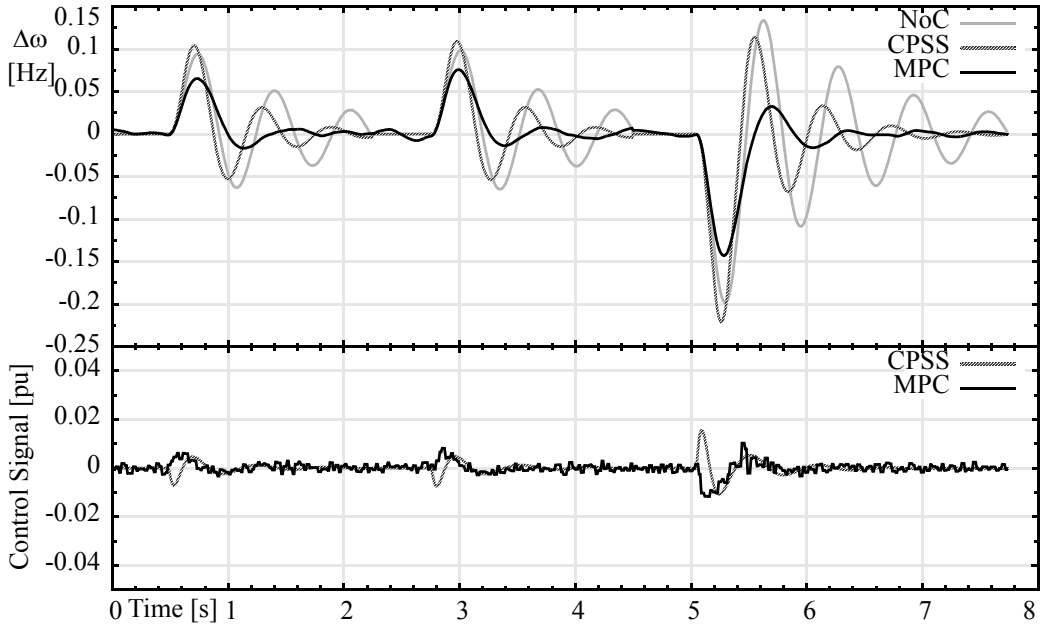


Figure 8-19 Voltage reference step changes in light load test

8.3.5 Leading Power Factor Test

In some situations it is required that the generator supplies a reactive load. For example, to absorb a capacitive load it may be necessary that the generator operate with a leading power factor. In this situation the stability margin of the generator is reduced. Therefore, it is vital to demonstrate that the controller is able to guarantee stable operation of the generator under the leading power factor conditions.

When the generator is operating in a single-machine infinite-bus system under leading power factor condition, $P = 0.5pu$ and $pf = 0.9lead$, a 0.15 pu step increase and

decrease in input mechanical torque is applied at 1 s and 5 s, respectively. The generator's rotational speed deviation with the control action for this test is shown in Figure 8-20.

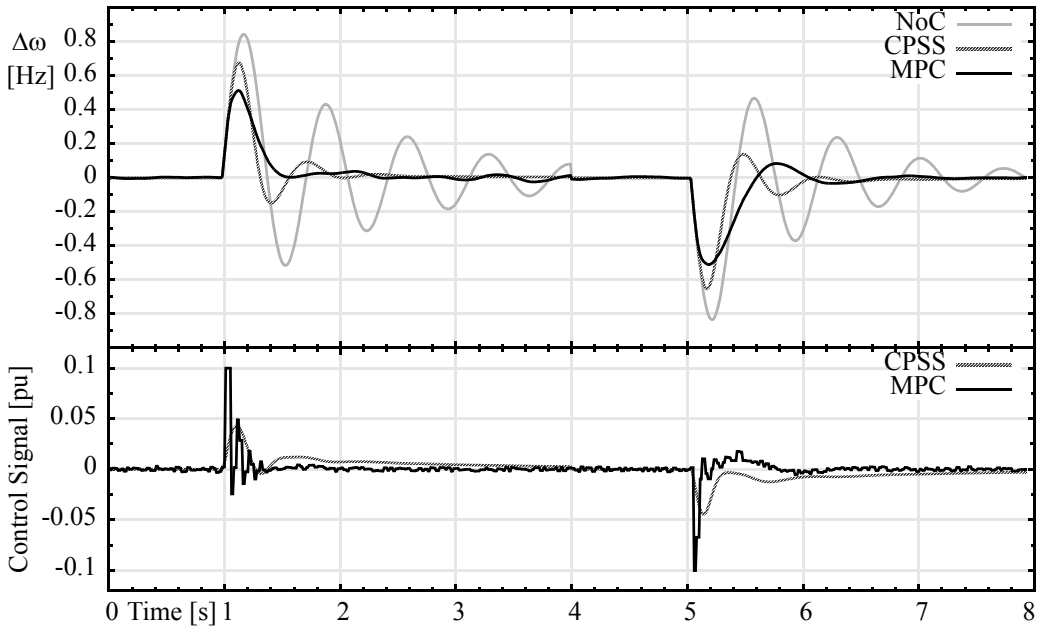


Figure 8-20 Response to a +/- 0.15 pu step change of torque in lead load test

Under the same load conditions, $P = 0.5pu$, $pf = 0.9lead$ and $V_t = 0.94pu$ terminal voltage, -0.015 pu, -0.015 pu and 0.03 pu step change in reference voltage is applied at 0.5 s, 2.8 s and 5 s respectively. The generating unit's rotational speed deviation with control action, is shown in Figure 8-21.

The results presented in Figure 8-20 and Figure 8-21 demonstrate that the oscillations are damped out by the H_∞ MPC with more success than with the CPSS, and that the H_∞ MPC can operate under leading power factor operating conditions.

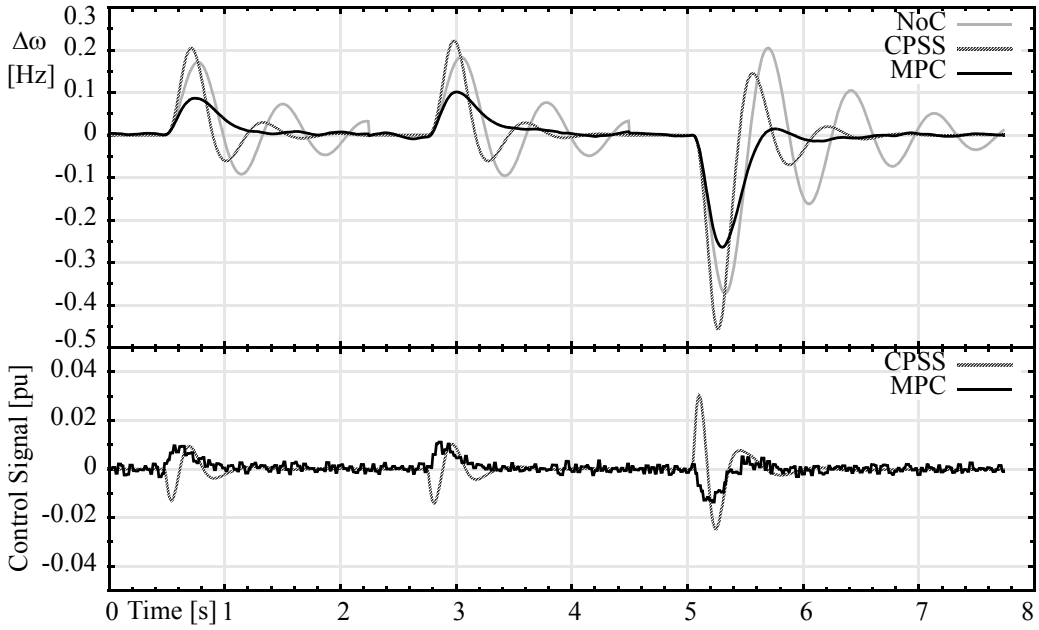


Figure 8-21 Voltage reference step changes in a lead load test

8.3.6 Transient Stability Test

Transient stability studies are aimed at determining if the system will remain in synchronism following major disturbances such as transmission system fault, sudden load changes, loss of generating units, or line switching [126]. For this test, the generating unit is operated under overload conditions, with $P = 1.1pu$ and $pf = 0.95lag$. Figure 8-22 illustrates the response of the power system to a three phase to ground fault test at the generator bus that is applied at 1 s and is cleared after 100 ms.

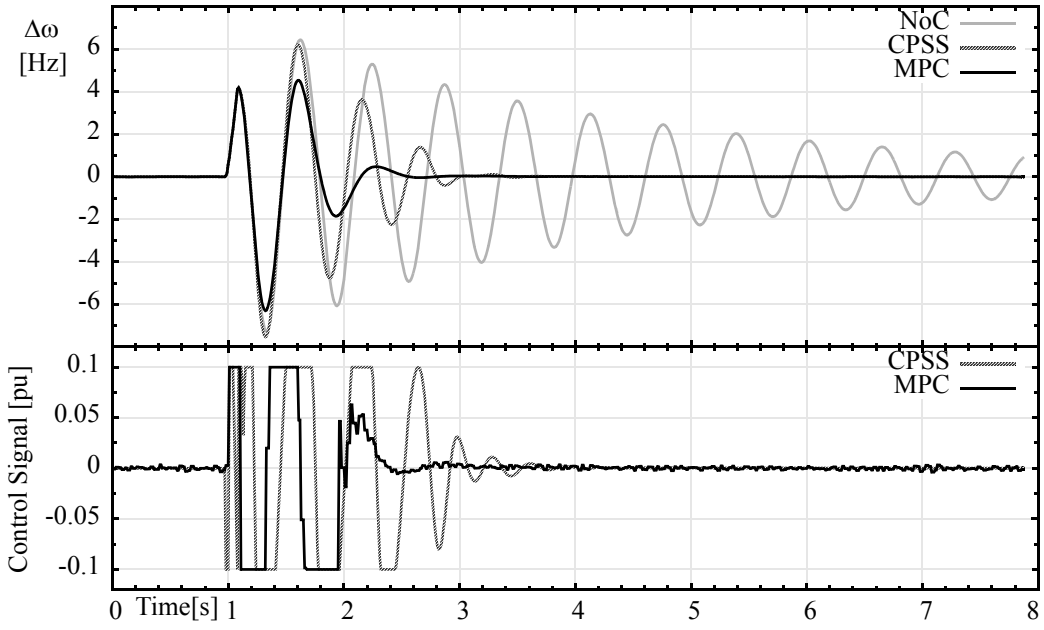


Figure 8-22 Bus three phase to ground fault test in overload condition

Test results demonstrate the superior behavior of the H_∞ MPC. This controller minimizes the peak of the generator's rotational speed deviation after the fault and helps the system return to its normal operating condition faster than the CPSS.

8.4 Summary

A qualitative analysis of the H_∞ MPC as power system stabilizer in a single-machine infinite-bus system is presented in Figures 8-13 through 8-22, and compared to the widely used CPSS. Simulation results verify that the H_∞ MPC has very good damping characteristics for disturbances under different operating conditions of the generating unit. For example, the machine settles to the new operating condition with a much smaller over-

shoot and oscillation amplitude than with the CPSS, therefore, improving the dynamic stability properties of the generator in the power system.

The robustness of the H_∞ controller, with respect to modelling errors and disturbance variations, stems from the fact that the maximum gain is minimized over all possible disturbances. Since no assumption is made about the disturbances during the definition of the H_∞ control, and all the conceivable disturbances have to be accommodated, the H_∞ control may be over conservative [20]. This over conservative behavior can be observed in Figure 8-13, Figure 8-18 and Figure 8-20, where the effects of disturbances with highest magnitudes are reduced while the small oscillations are left uncontrolled. This explains the effort of some researchers, whose aim is to develop a control algorithm which minimizes the mixed H_∞ / H_2 norm of the objective function.

9 Multi-Machine Power System

In the previous chapter, the results of the simulation studies demonstrated the H_∞ MPC controller properties as a power system stabilizer in a single-machine infinite-bus power system environment. In this chapter the performance of the H_∞ MPC controller is examined in a more complex test environment. Specifically, the implementation of the H_∞ MPC in a multi-machine power system is investigated. The power system simulation package, used in this research, is identical to that used in a single-machine infinite-bus power system simulation.

9.1 Introduction

Power system dynamics, with its transient response to a large disturbance, are characterized by multi-mode oscillation phenomena, with frequencies typically in the range of 0.1 to 2.0 Hz. Multi-mode oscillations emerge in multi-machine power systems, in which the interconnected generating units have quite different inertia and they are weakly connected by transmission lines. These oscillations are generally analyzed in three main oscillation modes; local, inter-area and inter-machine modes. Depending upon their location in the system, some generators participate in only one oscillation mode, while others participate in more than one mode [123].

Depending upon the origin of the oscillations, the multi-mode oscillations can be divided into three groups [120][129]:

- Local Mode is characterized by the oscillation frequencies in the range of 0.8 to 2 Hz. These oscillations are caused by the rotational speed deviation of a single generator connected to the infinite-bus, formed by the combined equivalent rotational speed of the interconnected power system.
- Inter-Machine Mode of oscillation is characterized by frequencies in the range of 0.3 to 1 Hz. This type of oscillation originates when the generators in a closely coupled group swing relative to each other. The PSS has no control over this type of oscillation.
- Inter-Area Mode of oscillation is characterized by frequency in the range of 0.1 to 0.7 Hz. These type of oscillations originate from a coherent group of generators in one area swinging relative to a number of other coherent groups of generators in other areas.

These multi-mode oscillations can be modelled by a five-machine power system.

9.2 Five-Machine Power Model

The interconnection scheme of a multi-machine power system model is shown in Figure 9-1. This model consists of five generator units. Generators G_1, G_2 and G_4 have much larger capacities than G_3 and G_5. The generators in this simulation are modelled with fifth-order non-linear differential equations, as given in Appendix B.1. All five generators are equipped with governors, exciters and automatic voltage regulators (AVR). The

parameters for the generators, AVRs, transmission lines and load operating conditions are given in Appendix B.

In the selected model, generators G_3, G_2 and G_5 form one area, while generators G_1 and G_4 form a second area. These two areas are linked through a tie line, which is connecting busses #6 and #7. Normally, each area serves its own load and it is almost fully loaded with a small load flow over the tie line [120].

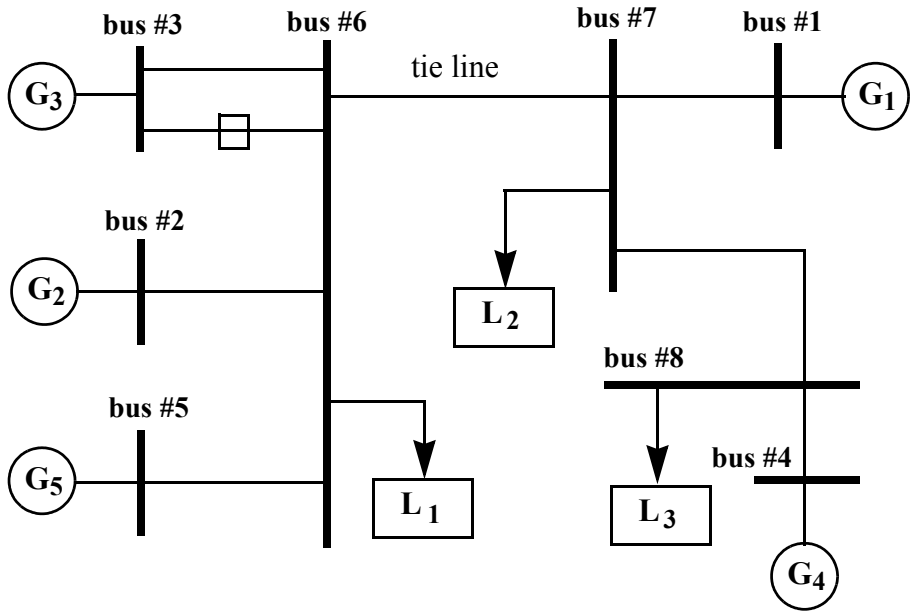


Figure 9-1 Network topology for a five-machine power system

Generator G_3 is connected to bus #6 through a double transmission line, which enables different fault tests on the second line. In Figure 9-1 a square indicates the second line, where the fault test takes place.

When the system is disturbed, multi-mode oscillations arise because of the different size generators and network configurations. The rotational speed difference between G_3

and G_2 exhibits mainly local mode oscillations, while the speed difference between G_2 and G_1 shows the inter-area mode oscillations. Both local and inter-area oscillations exist in the rotational speed difference between G_1 and G_3 [129].

If the multi-machine power system model is disturbed, the different size generators and the selected connections between the generators and the loads simulate the multi-mode oscillations. These multi-mode oscillations are presented in Figure 9-2. In this test, the system operates under operating condition #1, as defined in Appendix B.2. Then, a 0.1 pu step decrease in the mechanical torque reference of G_3 is applied at 1 s, and after 7 s the system returns to its original operating condition.

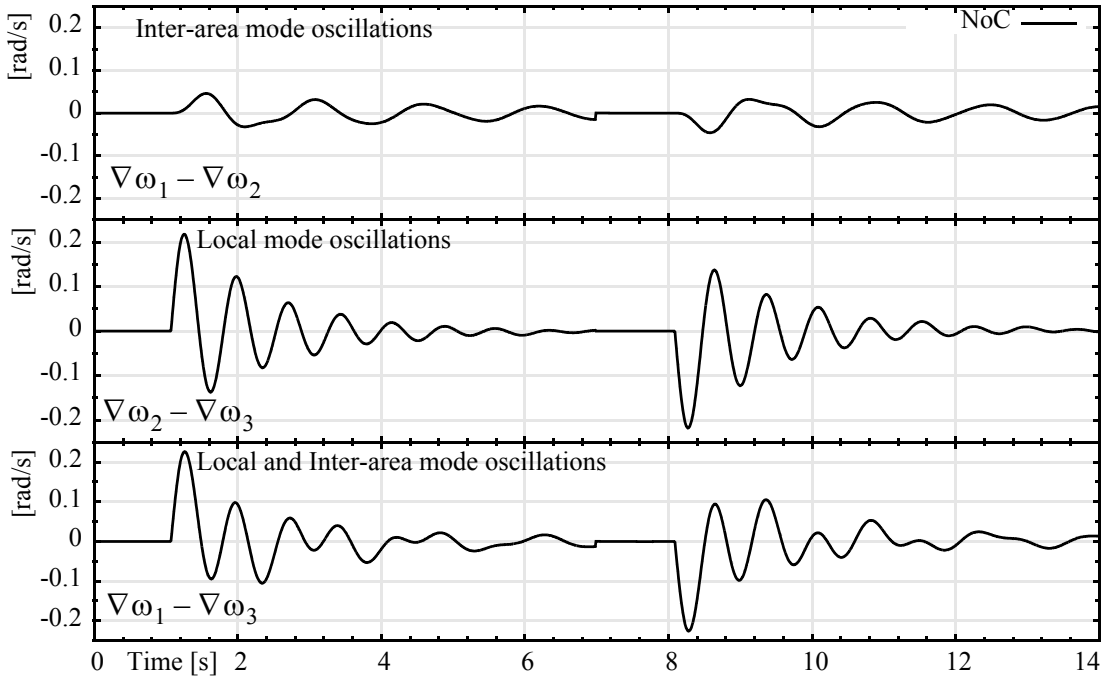


Figure 9-2 Multi-mode oscillations in a multi-machine power system

It can be observed from Figure 9-2 that the local mode oscillations arise from the speed difference between G_2 and G_3 at a frequency of about 1.3 Hz. The inter-area mode oscillations can be observed as a speed difference between G_1 and G_3 at a frequency of about 0.65 Hz, originating from the large difference of the generators' power used in the simulation model.

9.3 Simulation Studies for H_∞ MPC

The H_∞ MPC implemented in this test is the same one used in the single machine simulation with $T_S = 0.025s$ sample time. The speed deviation, $\Delta\omega$, used for the system state observation, contains a DC offset value since there is no infinite-bus in the system. A high-pass filter is implemented to remove the DC value before the speed deviation signal is used by the control system.

The high-pass filter introduces changes in the dynamics of the speed deviation, $\Delta\omega$, and it requires compensation by changing the weights in the H_∞ control

$D = 0.01 \cdot I^5$, $E = 0.001 \cdot 1^5$, $H = 0.2 \cdot \begin{bmatrix} I^5 \\ 0 \end{bmatrix}$ and $G = \begin{bmatrix} o^5 \\ 1 \end{bmatrix}$ where I^n is an $n \times n$ unity matrix and 1^n is an n length vector of *1s* and o^n is an n length vector of *0s*. The weights are kept constant during the simulation.

9.3.1 PSS on One Generator

During this test, in the multi-machine power system only one PSS is installed on the G_3 generator. When the system operates under operating condition #1, as defined in Appendix B.2, a 0.1 pu step decrease in the mechanical torque reference of G_3 occurs at 1 s and after 4 s the system returns to its original operating condition. The system response is shown in Figure 9-3. The H_∞ MPC installed on G_3 effectively damps the local mode oscillations, which arise as a rotational speed difference between the generators G_2 and G_3, $\nabla\omega_2 - \nabla\omega_3$. However, as expected, it has little influence on the inter-area mode oscillations, which arise as a rotational speed difference between the generators G_1 and G_3, $\nabla\omega_1 - \nabla\omega_3$. This is because the rated power of G_3 is much less than G_1 and G_2; and G_3 does not have enough power to control the inter-area mode oscillations [129].

For comparison, in a separate test, a CPSS, see Appendix A.3, with the following parameter set [130][120][129]

$$G_3: \quad K_s = 1.0 \qquad T_1 = T_3 = 0.3 \qquad T_2 = T_4 = 0.1 \qquad T_5 = 0.4 \qquad (9.1)$$

performs almost similar to the H_∞ MPC. Figure 9-3 presents both these test results and it also includes a response with no controller involved.

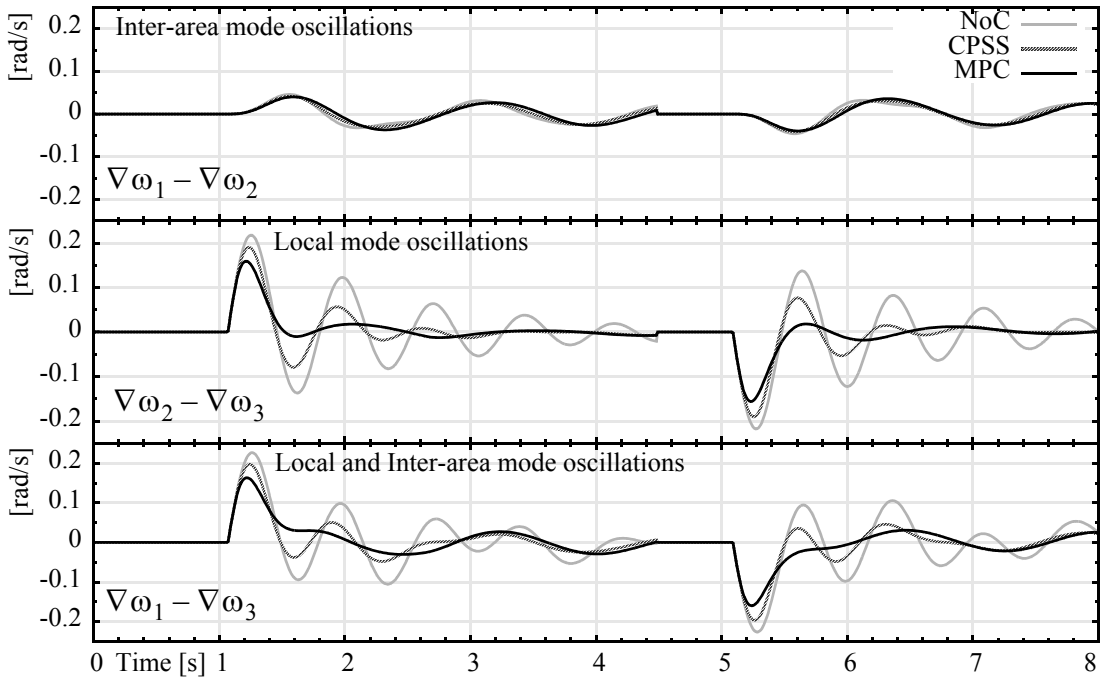


Figure 9-3 System response with PSS installed on G_3

9.3.2 PSS on Three Generators

In order to damp both local and inter-area mode oscillations, it is required that the PSS is installed on two additional generators G_1 and G_2, beside the already installed one on G_3. The H_∞ MPC is identical an all three units, while the CPSS requires different new parameters due to the difference in generator size [130]

$$G_1: \quad K_s = 0.4 \qquad T_1 = T_3 = 0.1 \qquad T_2 = T_4 = 0.08 \qquad T_5 = 0.3 \tag{9.2}$$

$$G_2: \quad K_s = 0.5 \qquad T_1 = T_3 = 0.1 \qquad T_2 = T_4 = 0.08 \qquad T_5 = 0.3\,. \tag{9.3}$$

This time the power system response is tested, when the system operates under the operating condition #1, as defined in Appendix B.2. A 0.1 pu step decrease in the mechanical torque reference of G_3 occurs at 1 s, and the system returns to its original working point after 4 s. The results are shown in Figure 9-4, which demonstrates that both modes of oscillations are damped effectively by the H_∞ MPC and by the CPSS.

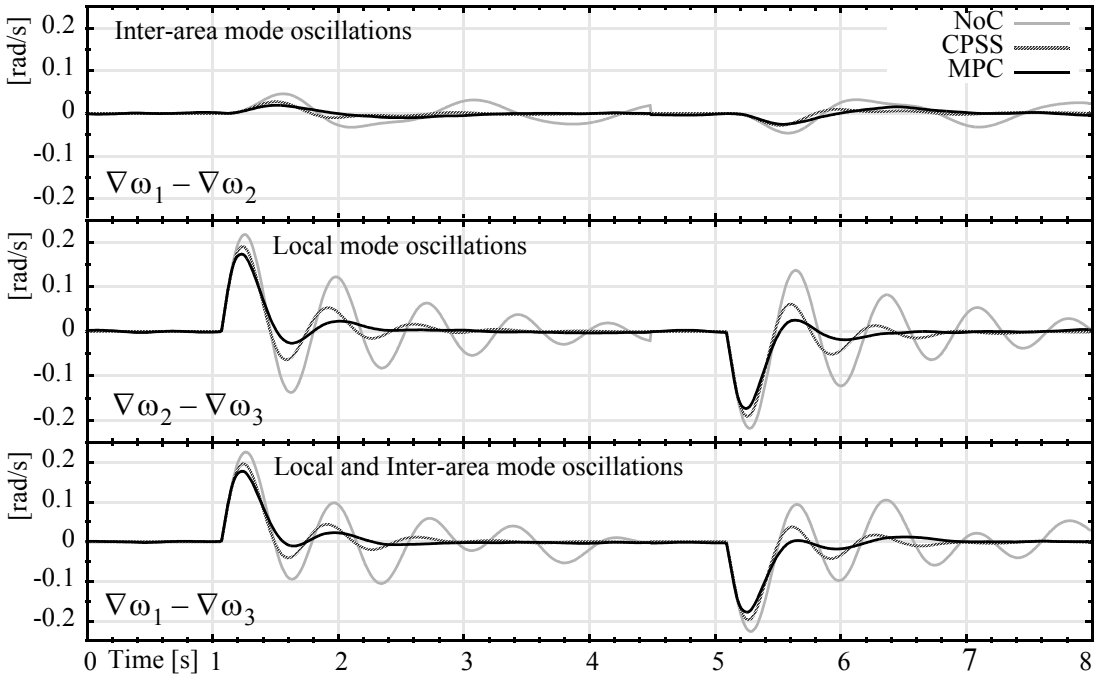


Figure 9-4 System response with PSS installed on G_1, G_2 and G_3

9.3.3 Mixed System with CPSS and H_∞ MPC

Since the power industry is widely using the CPSSs, any new PSS introduced to the system must be able to work together with the already installed stabilizers. This test is performed to demonstrate that in a mixed CPSS and H_∞ MPC environment, the two types of

PSSs can work co-operatively. In the multi-machine power system simulation, the H_∞ MPC is installed on G_1 and G_3, while the CPSSs are installed on G_2, G_4 and G_5. The parameters for CPSS on G_4 and G_5 generators are selected as [130]

$$G_4: \quad K_s = 0.5 \qquad T_1 = T_3 = 0.1 \qquad T_2 = T_4 = 0.08 \qquad T_5 = 0.3 \qquad (9.4)$$

$$G_5: \quad K_s = 1.0 \qquad T_1 = T_3 = 0.3 \qquad T_2 = T_4 = 0.1 \qquad T_5 = 0.4\,. \qquad (9.5)$$

When the system operates under the operating condition #1, as defined in Appendix B.2., the power system response is tested for a 0.1 pu step decrease in the mechanical torque reference of G_3, which occurs at 1 s. After 4 s the system returns to its original working point. The results in Figure 9-5 confirm that the two types of PSSs can work co-operatively.

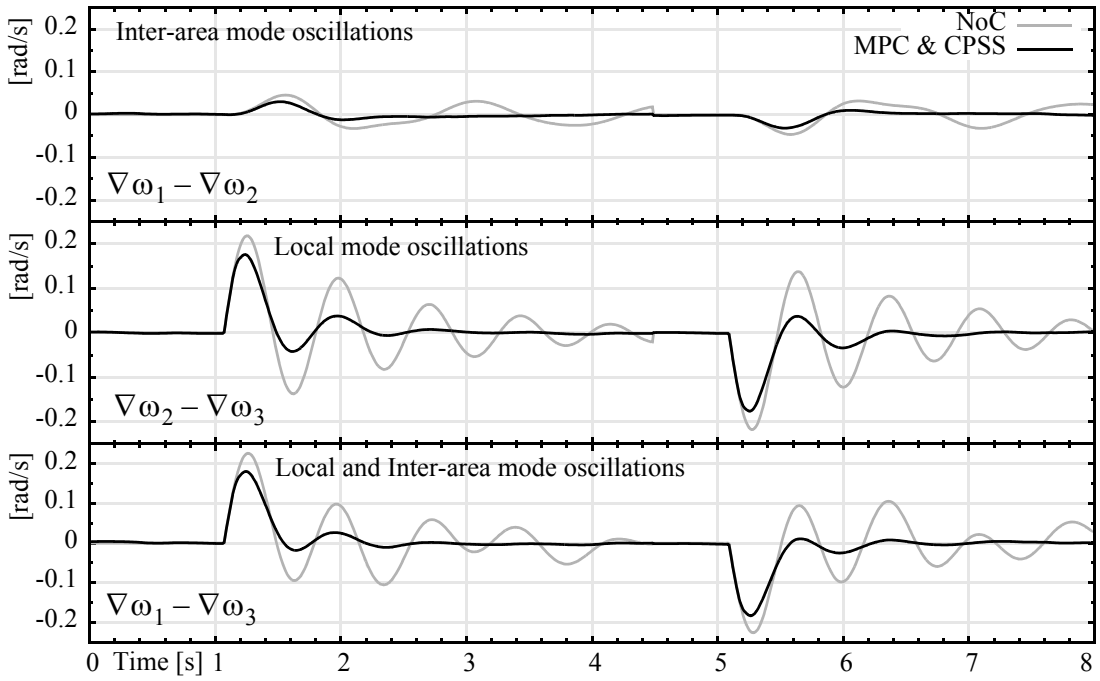


Figure 9-5 System response with a mixed CPSS and MPC installed

9.3.4 Three Phase to Ground Fault Test

With the power system operating under operating condition #1, as defined in Appendix B.2, a three phase to ground fault is applied at the middle of the transmission line between buses #3 and #6 at 1 s. The fault is cleared 100 ms later, when the faulty line is removed. Figure 9-6 shows the system response with no PSS, with the proposed H_∞ MPC, and with the CPSSs installed on G_1, G_2 and G_3. The results show that the H_∞ MPC minimizes the deviation of the generator's rotational speed after the fault.

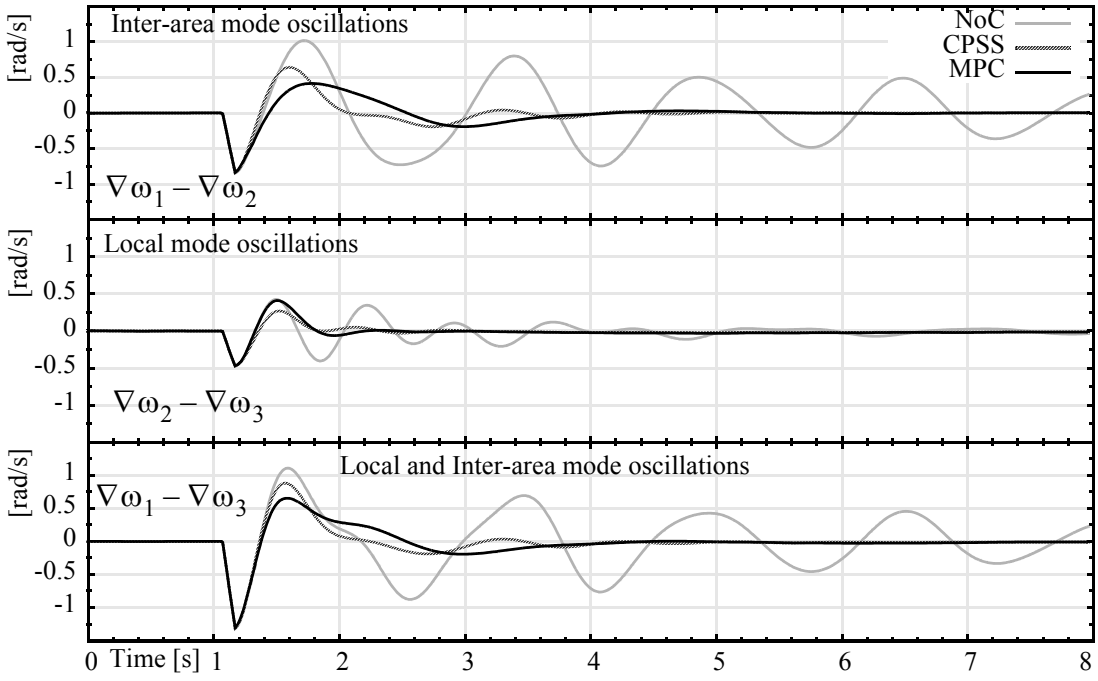


Figure 9-6 System response to three phase to ground fault test

9.3.5 New Operating Condition Test

The dynamic behavior of the power system changes with load and system operating conditions. To test the robustness of the proposed H_∞ MPC under other operating conditions, the operating point of the power system is changed to a new point as indicated in Appendix B.3.

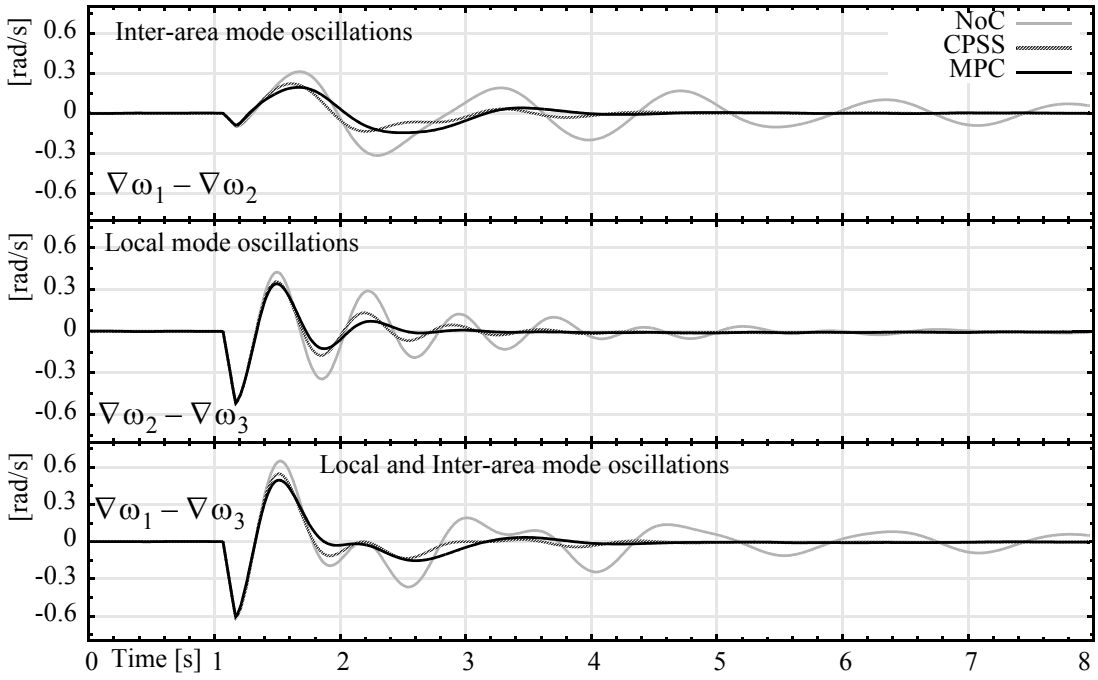


Figure 9-7 Three phase to ground fault test in a new operating condition

A three phase to ground fault is applied at the middle of the transmission line between buses #3 and #6 at 1 s, and cleared 100 ms later when the faulted line is removed. Figure 9-7 shows the system response with no PSS, with the proposed H_∞ MPC and with the

CPSSs installed on G_1, G_2 and G_3. Again, it is shown that the oscillations in the system with H_∞ MPC are more reduced than with CPSS.

9.4 Summary

In this Chapter, the effectiveness of an H_∞ MPC as a power system stabilizer is investigated in a multi-machine environment. The speed deviation of the generating unit is used as a signal to observe the generator's state. This signal contains a DC component, which has been successfully removed by a high-pass filter. During the simulation all of the off line selectable parameters of the H_∞ MPC are kept constant. The system changes are tracked by the online adaptation processes.

The simulation results indicate that the proposed H_∞ MPC not only performs well in damping multi-mode oscillations in the system but it also exhibits good adaptation and stable operation in all tested environments. It can be concluded that the proposed H_∞ MPC can produce satisfactory performance as a PSS both in single-machine and multi-machine power system environments. The performance of the proposed H_∞ MPC on a physical system is studied in the next part.

Part - IV Experimental Tests

An excellent way to gain an intuitive feel for control system design and performance is to conduct computer simulations. Conceptually, there is a short step from simulation to actual implementation, since the subroutines that are used on today's digital signal processors are very similar to those used for simulation facilitated by digital computers. However, a numerical system model can only approximate the dynamics of a system. There is always some unexpected dynamic behavior inherent in a system that is not accounted for by any given mathematical model. Consequently, it is necessary to do some laboratory and/or field tests to further assess the evidence from the computer simulation before the installation of the proposed control algorithm on an actual system occurs. This is especially true for a power system, in which the resulting damage could be very extensive. For this kind of system the solution is to build a scaled physical model of the target system, which is capable of emulating the behavior of the actual plant in the laboratory environment.

A real-time digital control system and the physical power system model for implementation and testing of the H_∞ MPC are described in this part.

10 Real-Time Test Environment

The next step in the progression of this thesis entails testing the proposed control strategy on a plant. In practice, new instrumentation and control schemes are tested on scaled physical models, that can simulate the behavior of the plant in the laboratory environment. Earlier studies have shown that most phenomena encountered in large power systems can be qualitatively duplicated with the small machines [131].

Implementation of the H_∞ MPC power system stabilizer with a real-time digital control system applied on a laboratory power system is described in this chapter. The proposed power system stabilizer setup is shown in Figure 10-1.

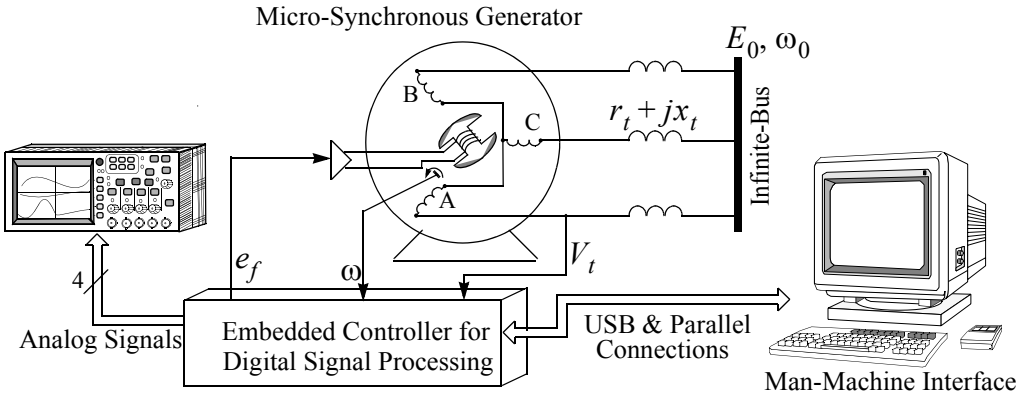


Figure 10-1 PSS development environment

The real-time test environment consists of two parts; physical model of a power system model and an embedded controller. The power system is modelled by a micro-synchronous generator connected to an infinite-bus through a transmission line. The

excitation of the micro-synchronous generator is controlled by the embedded controller using digital signal processing techniques. In order to observe the control system behavior in real time, a four-channel oscilloscope is utilized. The program for the embedded control system is developed on a personal computer. The personal computer is also used to control the program execution and the data collection.

In the following the functionality and characteristics of the elements of this power system model and the embedded controller are described.

10.1 Power System Model

The physical model of the power system to be simulated is available in the Power System Research Laboratory at the University of Calgary. It consists of a 3-phase 3kVA micro-synchronous generator connected to an infinite busbar through a double circuit transmission line simulator. An overall schematic diagram of this physical model is given in Figure 10-2. The major units of this model are: the turbine model, the generator model, the transmission line model, and the automatic voltage regulator.

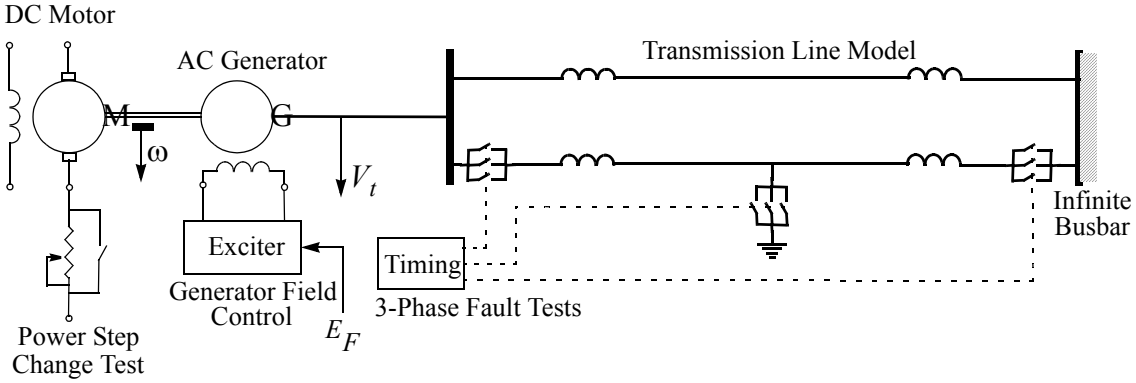


Figure 10-2 Laboratory power system configuration

10.1.1 Turbine Model

The mechanical energy source, in practice usually a turbine, is modelled by a 5.5 kW (7.5 h.p.) separately excited DC motor, made by Madsley's Ltd. in 1968. The technical parameters of this DC motor are: *220 V, 30 A, 1800 RPM, 7.5 h.p. (5592.75 W),* excitation *40/20/10 V.* The connection diagram with the measurement points is represented in Figure 10-3.

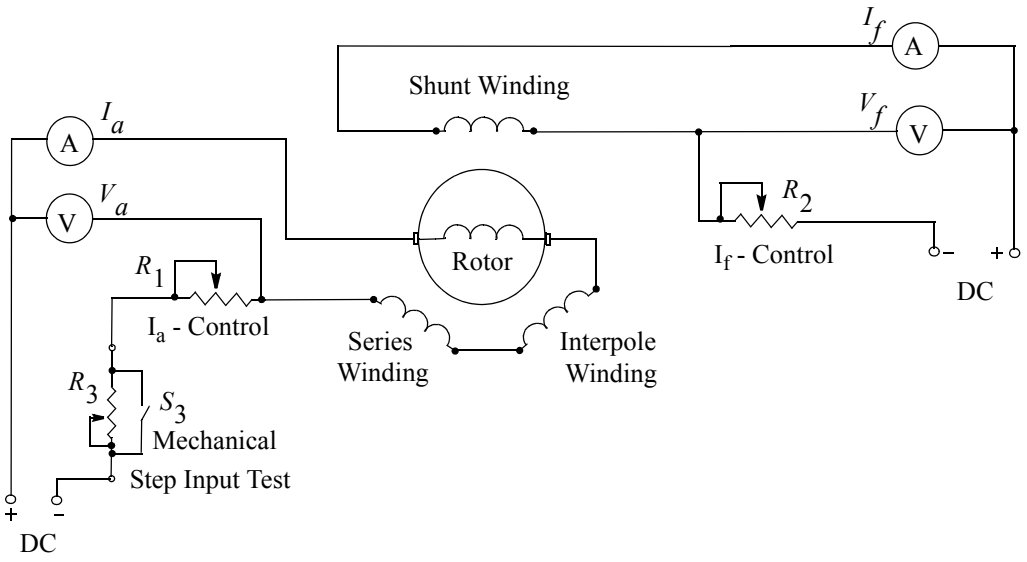


Figure 10-3 DC motor as a turbine model

The active electrical power, P_e, of the micro-synchronous generator, shown in Figure 10-4, can be set by varying the field current of the DC motor by the resistances R_1 and R_2. A large power step disturbance or input torque step, can be produced by using the switch S_3. The corresponding power step is adjusted by the resistor R_3.

10.1.2 Generator Model

A generator is modelled by a three phase 3 kVA, 220 V micro-synchronous generator, made by Madsley's Ltd. The technical parameters of this alternator are: 220/127 V, 7.9 A, 3 kW, 3 phase, 60 Hz with a power factor of 0.8 lag. The measurable parameters for the micro-synchronous generator are shown in Figure 10-4.

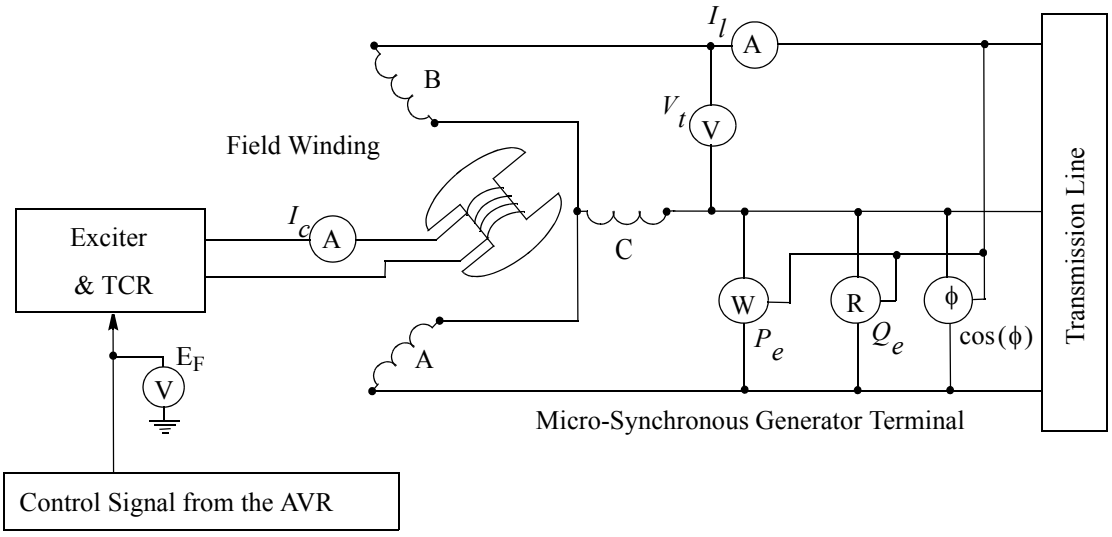


Figure 10-4 Micro-synchronous generator model

The field winding resistance of these synchronous machines, on per unit basis, is much higher than that of the large machines to be simulated. This results in a comparatively low field transient time constant $T_{do} = 0.765\ s$, where the transient time constant of the modern large machines is in the neighborhood of *5 s*, thus adversely affecting the performance of the machine under transient conditions. An electrical device called the Time Constant

Regulator (TCR) has been designed to reduce the effective field resistance and thereby alter T_{do} to the order of that required for the simulation of large machines. With the time constant regulator, the effective field time constant of the micro-synchronous generator can be increased up to *10 s* [132].

The voltage reference step disturbance can be simulated using this generator model, to conduct a dynamic performance study and evaluate the performance of the proposed power system stabilizer. The generator's terminal voltage, V_t, can be stepped up or down by changing the voltage reference setting of the AVR.

10.1.3 Transmission Line Model

The main objective of the transmission line is the power flow through an AC line, which is a function of the phase angle, the line end voltages, and the line-impedance [133].

The transmission line is modelled by a lumped element transmission line. The physical model consists of six π sections as presented in Figure 10-5, and gives a frequency response that is close to the actual transmission line response up to 500 Hz [134]. This

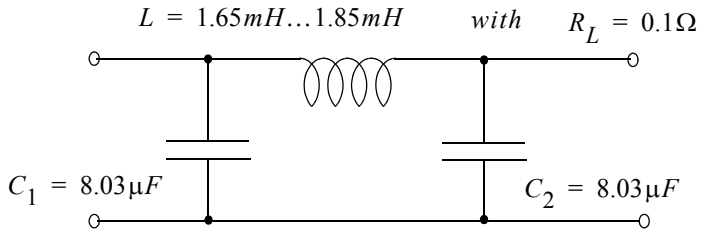


Figure 10-5 π section

model simulates the performance of a 300 km long, 500 kV, double circuit transmission line connected to a constant voltage bus (infinite busbar), as shown in Figure 10-1.

Large disturbances such as various single phase to ground, phase to phase and three phase to ground faults can be simulated using relays controlled by short circuit simulation logic, presented in Figure 10-1.

10.2 Embedded Controller

Real-time implementation of the proposed H_∞ MPC algorithm requires an embedded controller with higher computation power. Therefore, a powerful Digital Signal Processor (DSP) that existed at the time of the design was selected for building the controller. The DSPs developed by the Texas Instruments Incorporated were identified as most suitable.

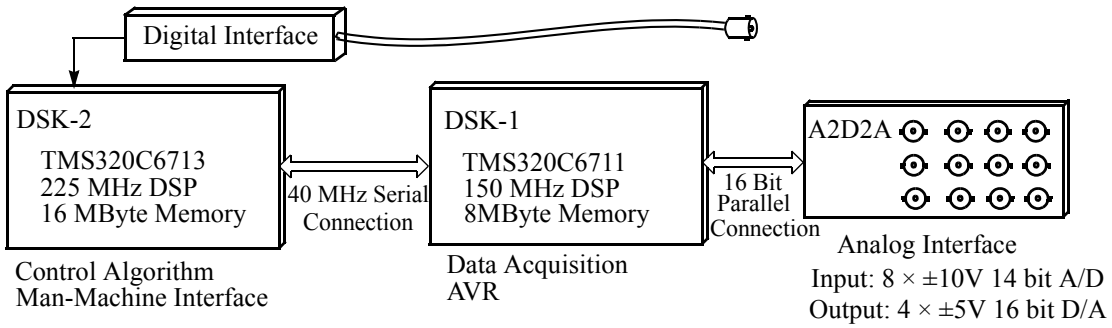


Figure 10-6 Embedded controller

The embedded control system consists of four cards: two computation cards and two interface cards, as shown in Figure 10-6. The four cards are mounted on a plexiglas panel. Two DSP Starter Kits (DSK), developed by Texas Instruments, are utilized for the compu-

tation. These DSK cards are the TMDS320006711 and TMDSDSK6713, containing high performance floating point DSPs TMS320C6711 and TMS320C6713, respectively. The technical description of the DSKs is provided in [135] and [136]. For software development additional technical literature [137 - 146] has been consulted.

The analog and digital interface cards were designed to satisfy the requirements of this research. The embedded system connection is presented in Figure 10-6. Their detailed description is provided in the following.

10.2.1 Analog Interface Card

In order to facilitate communication between the embedded controller and the environment, an analog interface card, named A2D2A, was developed. This card contains eight analog to digital (A/D) converter channels and four digital to analog (D/A) converter channels. For easy interfacing, the card uses common BNC coaxial connectors. The components that characterize the A2D2A are listed in Table 10-1, while its logical structure is presented in Figure 10-7.

Table 10-1 Characteristic components of the A2D2A card

Description	Part Number	Quantity	Manufacturer
A/D converter	AD7865AS-1	2	Analog Devices
D/A converter	DAC7644E-ND	1	Burr-Brown
Voltage Reference	LT1461DHS8-2.5	2	Linear Technology
Operational Amplifier	OPA2277UA	9	Burr-Brown
Buffer	BUF634U	4	Burr-Brown
Switching power supply	LT1533CS	3	Linear Technology
3.3V & 5V interface	SN74LVC4245APWR	3	Texas Instruments
Control logic	EMP7032AE	1	Altera Corporation

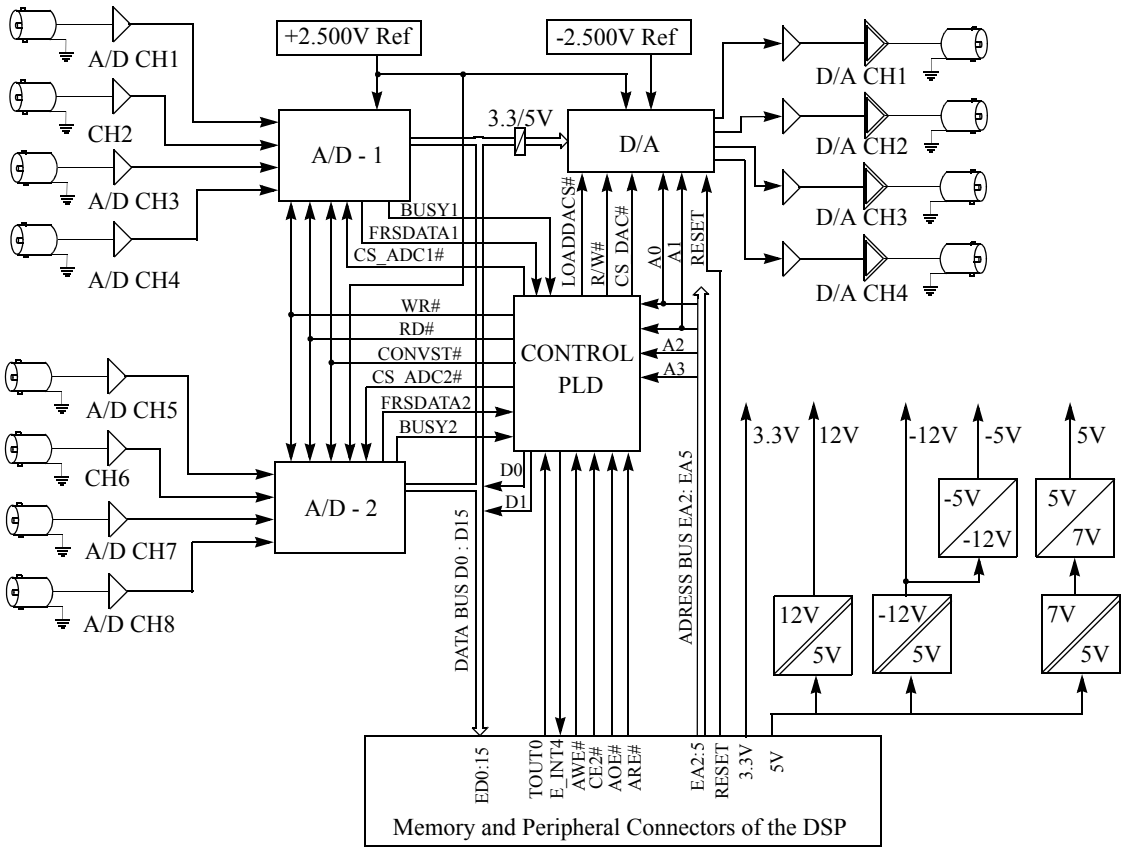


Figure 10-7 A2D2A signal diagram

In order to simplify the card's usage, the A2D2A is equipped with its own voltage converters for -12, -5, 5, and 12 V, which are required for functioning of devices.

The A2D2A board's conversion and access timing diagram is presented in Figure 10-8. Conversion is controlled by the Timer 0 output of the DSP, which generates the TOUT0 periodic pulses. The T_{TOUT0} period is controlled by the appropriate Timer 0 initialization, where the minimum period shall be selected for no less than 12 μs.

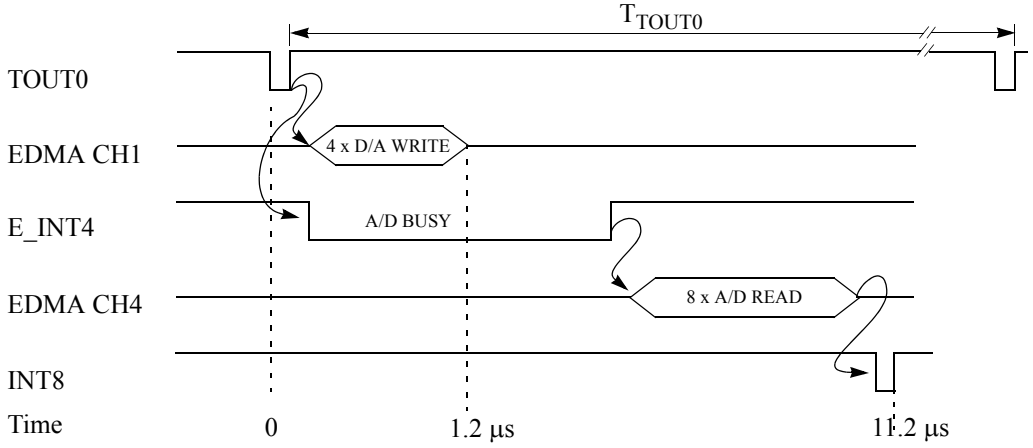


Figure 10-8 A2D2A conversion and access timing diagram

As presented in Figure 10-9, the TOUT0 pulse initiates the A/D conversion and it also starts a direct memory access (DMA) that updates the D/A converter registers. These processes occur simultaneously. The fourth D/A register will update within 1.2 μs from the start pulse. The A/D converter is designed to sample the eight channels concurrently to within a few nano seconds, thereby preserving the relative phase information between the input signals. At the end of A/D conversion, the E_INT4 interrupt signal initiates a second DMA access to the A2D2A board, in order to read the eight A/D registers. When the data transfer is complete, the DSP receives the INT8 interrupt, which signals that the converted data are ready for processing. In this design, the maximum conversion frequency can be selected up to 83 kHz.

In this research, the A/D and D/A converters are utilized to convert specifically at the frequency of 50.0 kHz, which corresponds to 20 μs T_{TOUT0} period. The DSP collects from all the eight channels sixteen samples of consecutive A/D conversion data and after calcu-

lating the average of the A/D conversions, it makes the data available every 0.32 ms to the rest of the processes.

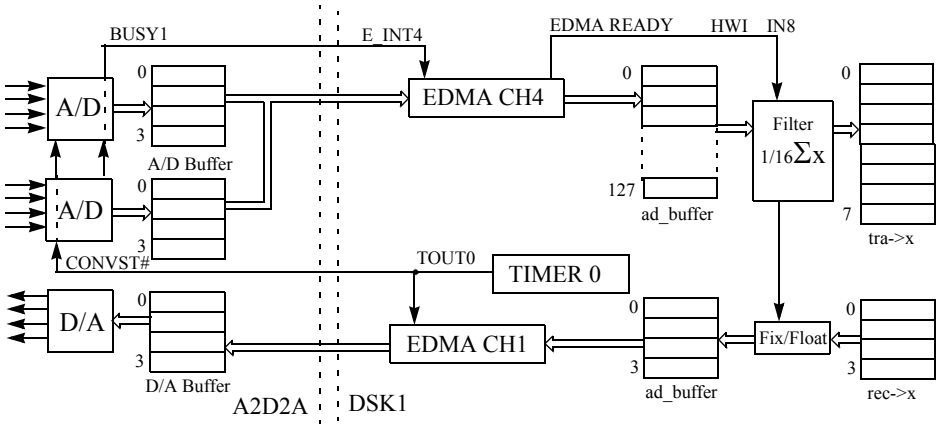


Figure 10-9 Data flow between A2D2A and the DSK1

10.2.2 Enhancement of A/D Conversion

In general, the data y_n from the A/D conversion contains some error due to the effects of the conversion process and the necessary signal conditioning prior to it. Usually this error can be modelled by an additive random signal ξ_n, superposed on the measured signal y_n^0 [147]

$$y_n = y_n^0 + \xi_n . \tag{10.1}$$

The accuracy of the A/D conversion improves if the signal to noise ratio is increased in the measurement. This is attained if multiple A/D conversions are performed and an average is calculated

$$\bar{y}_n = \frac{1}{N} \cdot \sum_{i=0}^{N} y_{n-i}. \tag{10.2}$$

For this procedure to be legitimate, the changes in the measured signal must be negligible on the length of the N sampling interval. Thus, $\bar{y}_n^0 \cong y_{n-1}^0 \ldots \cong y_{n-N}^0$ and

$$\bar{y}_n = \bar{y}_n^0 + \frac{1}{N} \cdot \sum_{i=1}^{N} \xi_{n-i} \tag{10.3}$$

where $\bar{y}_n^0$ is the average of the measured signal. The increase in the signal to noise ratio by the averaging process depends on the distribution of this random signal ξ_n. In a technical environment, this signal is usually a normally distributed random signal (white Gaussian noise) with zero mean. In the worst case scenario, this signal can be a uniformly distributed random signal. It is important to stress that if the mean of the random signal ξ_n is other than zero, the averaging process will not improve the measurements; rather, it will emphasize the bias of the nonzero mean.

- When the random signal $\xi_n \sim N(0, \sigma^2)$ is normally distributed with zero mean and variance σ^2, then the noise reduction in the A/D conversion data, obtained by averaging, is [148]

$$\bar{\xi}_n = \frac{1}{N} \cdot \sum_{i=1}^{N} \xi_{n-i} \text{ where } \bar{\xi}_n \sim N\left(0, \frac{1}{N}\sigma^2\right). \tag{10.4}$$

Even though the averaged signal is also normally distributed with a zero mean, there is a $1/N$ reduction in the noise variance, which corresponds to a $\sqrt{1/N}$ reduction of the standard deviation of the random signal. Consequently, the calculated average of 16 converted samples will improve the measurement accuracy by reducing four times the Gaussian noise in the measured data. In this way, the measurement accuracy improves by an extra two least significant valid bits.

- When the random signal $\xi_n \sim U(-\omega, \omega)$ is uniformly distributed in the interval $-\omega \le \xi_n \le \omega$, the corresponding variance is $\sigma^2 = \omega^2/3$ [149]. It is well known that the distribution of the sum of N uniformly distributed random signals rapidly approaches normal distribution, as N increases [150]. Accordingly,

$$\bar{\xi}_n = \frac{1}{N} \cdot \sum_{i=1}^{N} \xi_{n-i} \text{ where } \bar{\xi}_n \sim N\left(0, \frac{1}{N}(N\sigma^2)\right). \tag{10.5}$$

The variance of the sum is linearly increasing by the number of summing elements. This means that if the random signal $\xi_n \sim U(-\omega, \omega)$ is uniformly distributed, the signal accuracy cannot be improved by the averaging process. Fortunately, very few technical systems have uniformly distributed contaminating signals.

10.2.3 Digital Interface Card

The digital interface card is used to connect the pulses, generated by the generator's rotation, to an interrupt line on the DSP. The full rotation of the generator produces four pulses. The rotational speed of the generator is 1800 revolution per minute, which corre-

sponds to 30 Hz of rotation. If the pulses were to interrupt the DSP at 120 Hz frequency, the DSP would overload. In order to avoid this overload, the pulses are divided by four on the digital interface card. Also, to alleviate accidents from high voltage sparks which can harm the DSP, the digital interface card has an optocoupler signal isolation built in it, as presented in Figure 10-10. The function of the LED on the digital interface card is to alert if the signal connection has been severed. During operation, a uniform blinking light signals normal conditions.

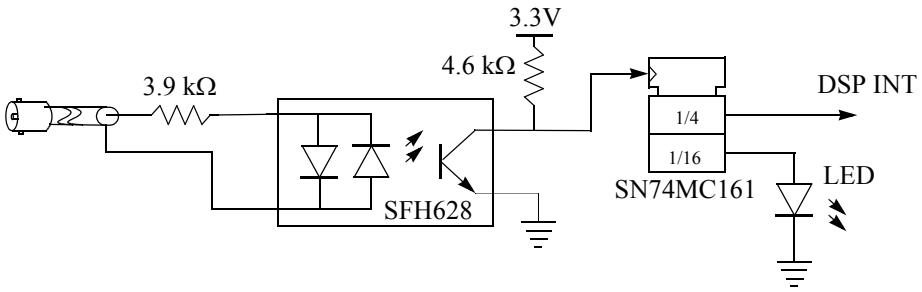


Figure 10-10 Digital interface signal diagram

The synchronous generator, as part of the power system, is observed through its rotational speed deviation, $\Delta\omega$, as presented in Figure 10-11. The rotational speed, or more precisely, the rotational speed deviation of the synchronous generator is observed by measuring the time length of one full rotation. In order to measure the oscillations in the rotational speed, the DSP must calculate the difference between the measured rotational time and the moving median of the last 25 rotational times. Using the median as a reference rotational time assures that the rotational speed deviation does not contain an unwanted offset from zero.

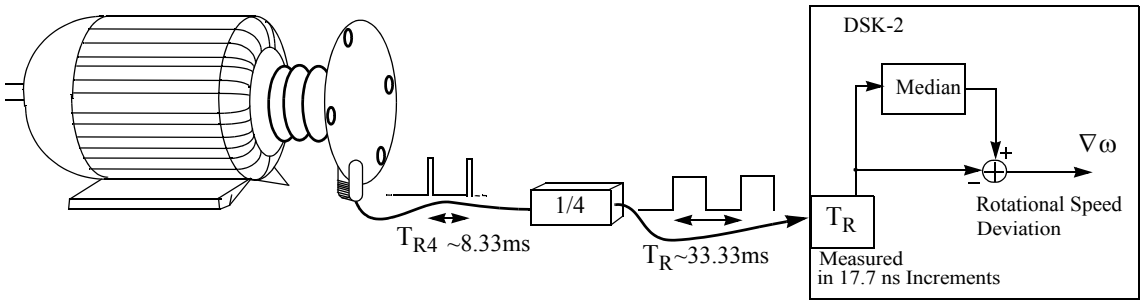


Figure 10-11 Rotational speed deviation estimation

10.2.4 Embedded Software

The embedded software is divided between the two DSKs in such a way that the Exciter-AVR function with the analog interface handling is allocated to the DSK-1, while the control algorithm, containing the H_∞ MPC or CPSS, with the digital interface handling is allocated to the DSK-2. Between the two DSKs a serial link is established which continuously exchanges two blocks of data. Each block contains 16 data. The real-time control application is primarily interrupt driven, with a basic structure shown in Figures 10-12 and 10-13 for DSK-1 and DSK-2 respectively.

The primary function of the software on DSK-1 board is to handle the analog data communications and the AVR. The timing of the software execution is driven by two periodic interrupts of 20 μs and 0.32 ms. Every 20 μs the software reads the data registers of the A/D converters and updates the D/A data registers on the A2D2A board. From the A/D readings of all eight analog input channels, the software collects 16 samples during a 0.32 ms long interval. The A/D Data Filter calculates the average of these 16 A/D samples

that are used in subsequent signal processing. This way the Gaussian noise in the measured data is reduced four times.

The primary control loop for the generator's voltage stability is the AVR, which also executes with 0.320 ms sampling time. The equations for AVR are provided in Section 10.3.1.

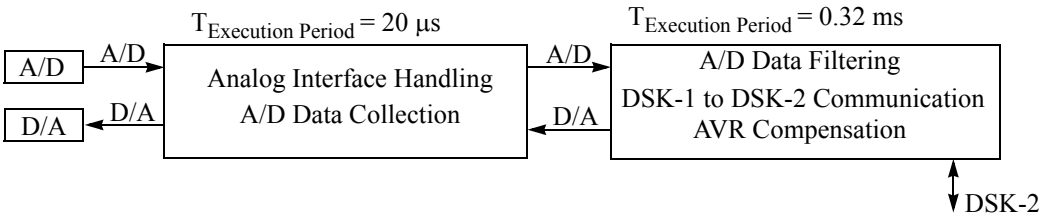


Figure 10-12 DSK-1 program allocation and execution

A synchronous serial connection line is used for communication between DSK-1 and DSK-2 boards. Every 4 μs the block of the 16 data is exchanged between the two DSKs.

DSK-2 contains a higher level of calculations. Its execution period is much longer than that of DSK-1. The execution cycle of the primary real-time application is 33.33 ms, which includes the H_∞ control action calculations or the CPSS and the system identifications by different techniques. The secondary applications, which include the expert system for Proposed (PRO) model selection, are executed at 1 s interval.

If a new PRO model is selected, a new set of weights are calculated for the real time H_∞ controller using the H_∞ optimization algorithm in a background calculation, as defined in sections 6.2.2 through 6.2.4.

The real-time control application program is interrupt driven. To measure the period of a rotation, it uses the external interrupt from the generator's rotation, as presented in Fig-

ure 10-11. Due to this particular interrupt driven structure, the control action can be immediately calculated when the measurements from the rotational speed deviation are available. Consequently, the delay is minimized between sampling and control. The calculation of the system identification algorithms takes place in the same real time loop of 33.33 ms.

In this research, seven different system identification techniques are concurrently calculating the system model. Every second a new interrupt is generated which invokes the expert system in order to make a decision for the PRO model. The expert system has eight possible choices, either it can keep the PRO model or it can replace it with one of the seven possible models resulting from the seven system identification techniques.

If a new model is proposed for PRO, the H_∞ optimization algorithm starts to execute in the background. The reason for this background action is that this part of the algorithm requires a longer than the 33.33 ms interval for the calculations. When the new parameters are available, the old weights in the H_∞ controller are replaced.

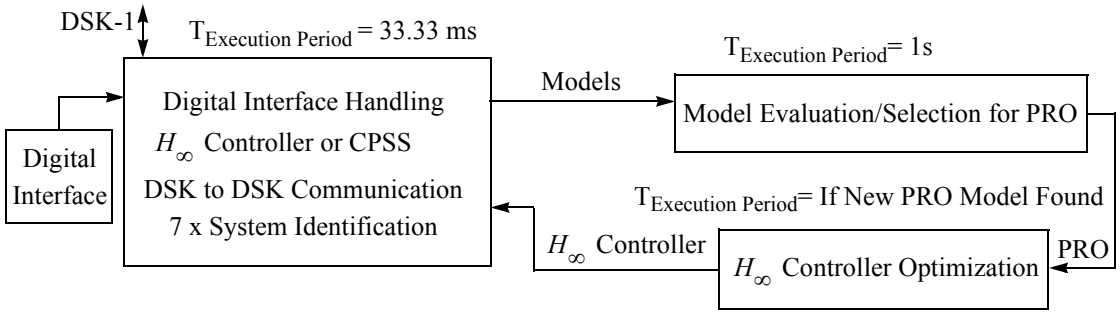


Figure 10-13 DSK-2 program allocation and execution

10.3 Power System Stabilizer Implementation

As described above, the real-time control environment of a power system stabilizer is based on an embedded controller. The embedded controller is a state of the art computation unit containing two floating point digital signal processors with the required interfaces. The implementation and the test environment are illustrated in Figure 10-14.

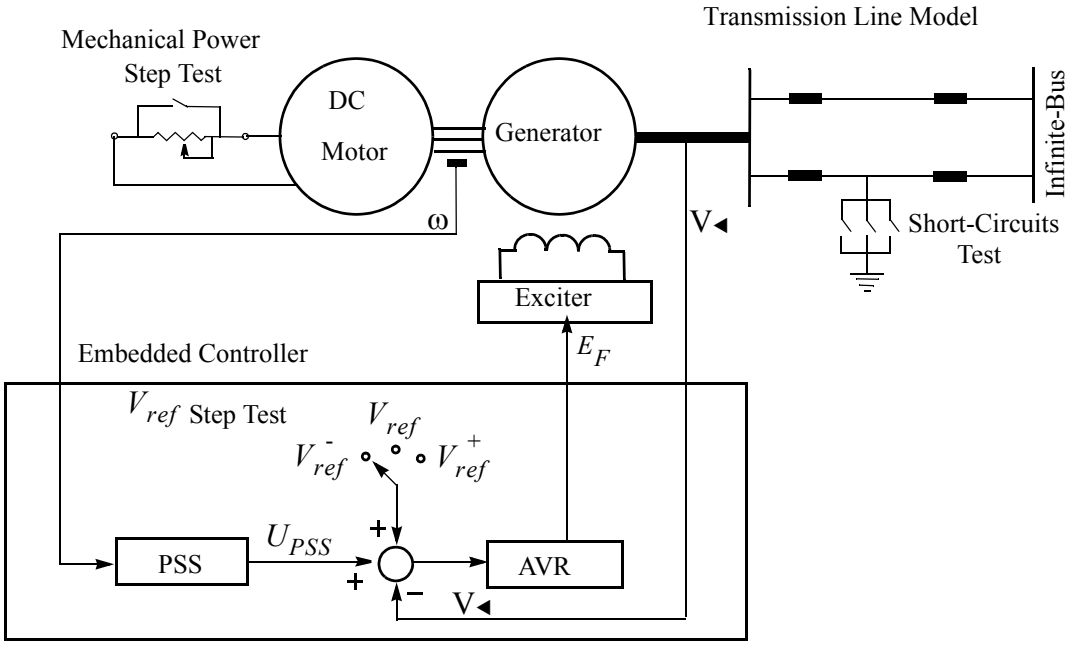


Figure 10-14 Tests on power system model

10.3.1 Excitation System

The main purpose of the excitation system is to feed the field winding of the synchronous machine with direct current so that the main flux in the rotor is generated. Further, the terminal voltage of the synchronous machine is controlled by the excitation system, which also performs a number of protection and control tasks [151].

The terminal voltage, V_t, and rotation speed, ω, of a synchronous generator are two important quantities on which the operation of the power system depends. Special means are necessary to control them if the best operating conditions are to be obtained. As shown in Figure 10-15, terminal voltage, V_t, and rotation speed, ω, can be used to control the synchronous generator's magnetic field.

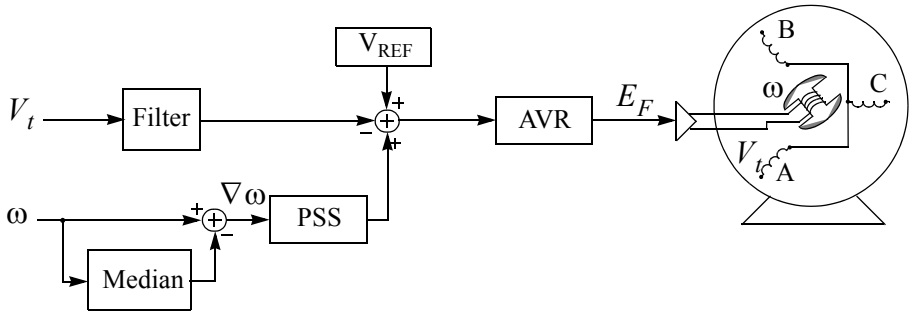


Figure 10-15 Excitation system with AVR and PSS

According to the *North American Electric Reliability Council* (NERC) requirement definitions as it is stipulated in the NERC Planning Standards, all synchronous generators connected to the interconnected transmission systems shall be operated with their excitation system in the automatic voltage control mode [152]. The terminal voltage of a gener-

ator operating by itself can be wholly controlled by an Automatic Voltage Regulator (AVR) actuated by voltage feedback. When the generator is connected to a large power system (infinite busbar), its voltage and rotation speed are largely determined by the voltage and frequency of the system. The main effect of the automatic voltage regulator is to control the reactive power, Q_e, of the generator [153].

Automatic voltage regulators (AVRs) shall be continuously acting solid state analog or digital. Tuning should be in accordance with the *NERC Planning Standard Guide IIC G8* cited below [152]

- Generator voltage regulators, to the extent practical, should be tuned for fast response to step changes in terminal voltage or voltage reference. It is preferable to run the step change in voltage tests with the generator not connected to the system so as to eliminate the system effects on the generator voltage. Terminal voltage overshoot should generally not exceed 10% for an open circuit step change in a voltage test.

The AVR and exciter used in the experimental studies is a discrete version of the analog AVR defined in the IEEE Standard P421.5/D15, Type ST1A, shown in Figure A-2.

The input signal for the controller is the voltage error, V_I, from the compensation equipment. The stabilizing feedback, V_F, is subtracted and the signal from the PSS, U_{PSS}, is added. Both these signals vanish in steady state. The controller is mainly described by the dominating time constant, T_A, and the amplification, K_A. The limits represent saturation effects or limitations of the control signal. The time constants, T_C, T_{C1}, T_B, and T_{B1} are used to model internal time constants in the controller. These are often

small and can then usually be neglected [151]. The output signal, E_F, from the AVR controls the exciter.

In the IEEE Standard the transfer function of the AVR is defined in the complex domain from which, in practical applications the best discrete transformation can be obtained using the bilinear z-transformation [154]

$$s = \frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}. \tag{10.6}$$

The V_{C1} to V_C filter numerically can be approximated by the discrete transfer function

$$\frac{V_C}{V_K} = \frac{1}{1 + \left(\frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) T_R} = \frac{\frac{T_s}{T_s + 2T_R} + \frac{T_s}{T_s + 2T_R} \cdot z^{-1}}{1 + \frac{T_s - 2T_R}{T_s + 2T_R} \cdot z^{-1}}. \tag{10.7}$$

The lead-lag part numerical approximation using the bilinear z-transformation is

$$\frac{(1 + sT_C)(1 + sT_{C1})}{(1 + sT_B)(1 + sT_{B1})} \rightarrow \tag{10.8}$$

$$\frac{\left(1 + \left(\frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) T_C\right)\left(1 + \left(\frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) T_{C1}\right)}{\left(1 + \left(\frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) T_B\right)\left(1 + \left(\frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) T_{B1}\right)} = \frac{N_0 + N_1 \cdot z^{-1} + N_2 \cdot z^{-2}}{1 + D_1 \cdot z^{-1} + D_2 \cdot z^{-2}}$$

where $N_0 = \dfrac{1+\dfrac{2}{T_S}(T_C+T_{C1})+\dfrac{4T_CT_{C1}}{(T_S)^2}}{1+\dfrac{2(T_B+T_{B1})}{T_S}+\dfrac{4T_BT_{B1}}{(T_S)^2}}$, $N_1 = \dfrac{2-8\dfrac{T_CT_{C1}}{(T_S)^2}}{1+\dfrac{2(T_B+T_{B1})}{T_S}+\dfrac{4T_BT_{B1}}{(T_S)^2}}$,

$N_2 = \dfrac{1-\dfrac{2}{T_S}(T_C+T_{C1})+\dfrac{4T_CT_{C1}}{(T_S)^2}}{1+\dfrac{2(T_B+T_{B1})}{T_S}+\dfrac{4T_BT_{B1}}{(T_S)^2}}$, $D_1 = \dfrac{2-8\dfrac{T_BT_{B1}}{(T_S)^2}}{1+\dfrac{2(T_B+T_{B1})}{T_S}+\dfrac{4T_BT_{B1}}{(T_S)^2}}$, and

$D_2 = \dfrac{1-\dfrac{2}{T_S}(T_B+T_{B1})+\dfrac{4T_BT_{B1}}{(T_S)^2}}{1+\dfrac{2(T_B+T_{B1})}{T_S}+\dfrac{4T_BT_{B1}}{(T_S)^2}}$.

The output filter is approximated as

$$\frac{K_A}{1+sT_A} \rightarrow \frac{K_A}{1+\left(\dfrac{2}{T_s}\cdot\dfrac{1-z^{-1}}{1+z^{-1}}\right)T_A} = \frac{\dfrac{K_A}{1+2T_A/T_s}+\dfrac{K_A}{1+2T_A/T_s}\cdot z^{-1}}{1+\dfrac{1-2T_A/T_s}{1+2T_A/T_s}\cdot z^{-1}} \tag{10.9}$$

and, finally, the derivative feedback filter is approximated as

$$\frac{sK_F}{1+sT_F} \rightarrow \frac{\left(\dfrac{2}{T_s}\cdot\dfrac{1-z^{-1}}{1+z^{-1}}\right)K_F}{1+\left(\dfrac{2}{T_s}\cdot\dfrac{1-z^{-1}}{1+z^{-1}}\right)T_F} = \frac{\dfrac{2K_F/T_s}{1+2T_F/T_s}-\dfrac{2K_F/T_s}{1+2T_F/T_s}\cdot z^{-1}}{1+\dfrac{1-2T_F/T_s}{1+2T_F/T_s}\cdot z^{-1}}. \tag{10.10}$$

Using the discretization defined in (10.7) through (10.10), the rest of the implementation of the digital AVR on the DSP follows the IEEE standard depicted in Figure A-2. The

parameters for the AVR are selected using the suggestion from [155] and fine-tuned afterwards. The selected parameters are presented in Table 10-2.

Table 10-2 AVR parameters used in experimental tests

$R_C = 0.0$	$X_C = 0.0$	$T_R = 0.0001$	$V_{T_{REF}} = 2.95$
$V_{I_{MIN}} = -5.0$	$V_{I_{MAX}} = 5.0$	$V_{UEL} = 0.0$	$T_s = 0.03333$
$T_B = 0.0$	$T_C = 0.0$	$T_{B1} = 0.0$	$T_{C1} = 0.0$
$T_A = 0.01$	$K_A = 12.0$	$V_{A_{MIN}} = -5.0$	$V_{A_{MAX}} = 5.0$
$I_{LR} = 0.0$	$K_{LR} = 0.0$	$V_{OEL} = 5.0$	$T_F = 1.0$
$K_F = 1.0$	$K_C = 0.080$	$V_{R_{MIN}} = -4.0$	$V_{R_{MAX}} = 4.0$

10.3.2 Conventional Power System Stabilizer (CPSS)

The proposed H_∞ MPC control algorithm is compared against the CPSS which is widely used in the power industry. In order to obtain credible test results, the CPSS was implemented in the same environment. The CPSS transfer function consists of a gain, low-pass filter, high-pass filter and a lead-lag compensator [156]

$$\frac{Upss(s)}{\nabla\omega(s)} = K_s \cdot \frac{1}{1+sT_6} \cdot \frac{T_5 s}{1+sT_5} \cdot \frac{(1+sT_1)(1+sT_3)}{(1+sT_2)(1+sT_4)} \tag{10.11}$$

where K_s is the stabilizer gain, T_6 is the low pass filter's time constant, T_5 is a high pass filter's time constant (wash-out filter), and $T_1 \ldots T_4$ are the time constants for lead-lag stages, which are configured as a double phase-lead compensator by selecting $T_1 > T_2$ and $T_3 > T_4$. The discrete transfer function is obtained substituting (10.6) into (10.11). The

discrete transfer function with simple variable changes is analogous to (10.7), (10.8) and (10.10).

The CPSS parameters are tuned using the tuning procedure described in [128] and [156-160]. The parameters for the CPSS algorithm are presented in Table 10-3. These parameters are the result of the fine tuning of the CPSS parameters suggested in [156], when the generator is operating at $P = 0.8pu$ and with a $pf = 0.95lag$. This operating point is selected because it is assumed to be the highest operating point of the power system model which is able to withstand without permanent failure disturbances caused by the CPSS misbehavior during tuning. If a much lower operating point were selected the generator's rotational speed deviation would be negligible with no control response from the CPSS.

The objective of the fine tuning of the CPSS parameters is to minimize the duration of the oscillations in generator's rotational speed deviation. For the tuning, a *0.15 pu* mechanical torque step change is used as an external disturbance. During the tuning process the generator's rotational speed deviation is observed on an oscilloscope.

Table 10-3 CPSS parameters used in experimental tests

$T_1 = 0.25$	$T_2 = 0.005$	$T_3 = 0.5$
$T_4 = 0.06$	$T_5 = 0.1$	$T_6 = 1.6$
$K_S = -0.2$	$T_s = 0.03333$	

The parameters of the CPSS are then kept unchanged for all the experimental tests.

11 Real-Time Tests

Result of experimental studies with the proposed H_∞ MPC as a power system stabilizer are presented in this chapter. Performance of the proposed H_∞ MPC has been investigated on a physical model of a single machine infinite bus power system described in Chapter 10, under different operating conditions and disturbances. For comparison, a digital version of the conventional power system stabilizer (CPSS) has been also implemented in the same environment and tested under the same conditions, see section 10.3.2.

All experimental data were collected through a human-machine interface and were saved on a disk for further analysis. The results of the data analysis are presented in this chapter. On the figures, the time axis has been adjusted so that the disturbances occur coincidentally, ensuring a consistent visual presentation.

11.1 System Identification of the Power System Model

To evaluate the proposed multiple model system identification in a stationary environment, the micro-synchronous generator was operated at power $P = 0.9$ p.u. with a power-factor $\text{pf} = 0.93$ lag. During this experiment, data is collected over a 20 minute time interval, while the system is undisturbed.

To model the power system, a fifth order ARMA model was selected. The system output with the ARMA model, $\hat{y}_n$, is predicted using

$$\hat{y}_n = \sum_{i=1}^{5} b_i u_{n-i} - \sum_{i=1}^{5} a_i y_{n-i} \tag{11.1}$$

where u_{n-i} and y_{n-i} are the measured system input and output, respectively. The a_i and b_i parameters are the result of the system identification process. They are presented in Figures 11-9 and 11-10. The powers of residuals, used to compare different techniques, are calculated as the difference between the predicted and measured system outputs

$$v_n^2 = (\hat{y}_n - y_n)^2 . \tag{11.2}$$

In Figures 11-1 through 11-6, the powers of residuals from the different system identification techniques are compared with that of the PRO model. This PRO model is proposed by the expert system. Values for the powers of residuals are averaged in these figures. These are obtained using the "*smooth bezier*", a built in function of the *Gnuplot 4.0,* plotting program.

11.1.1 Comparison of System Identifications

In this research the proposed approach to system identification is to implement multiple techniques with different behaviors concurrently, while an expert system selects the best model periodically. All system identification techniques used are parametrized in their unique ways to evoke different identification behaviors. The used parameters for the algorithms in this experiment are presented in Table 11-1. As a result, system identifications are characterized by different adaptation speeds and accuracy. The goal is to utilize this in various transient situations of a plant.

Table 11-1 Parameters in system identifications

Techniques	Parameters
RSL	$\lambda = 0.985$
LMS	$\mu = 0.7$
WCE	$\delta = 1.0$
KF	$R = 0.2\ \mathbf{I}^{qxq}$; $Q = 0.2\ \mathbf{I}^{mxm}$
FQR	$\lambda = 0.975$
IQR	$\lambda = 0.975$
QRL	$\nu = 0.9$

In Figures 11-1 through 11-7, lower residual power, v^2, shows a better system model. From these figures it is clear that not all system identification techniques perform adequately. However, that is not the expectation. The objective is that from all these techniques the expert system is able to select the best possible PRO model.

RLS algorithm. System identification by the RLS algorithm compared to the PRO, in Figure 11-1, shows a lag in initial convergence. After the first *100 s* the performance of the RLS and PRO are indistinguishable, which explains why the expert system selects so frequently the parameters from the RLS algorithm for the proposed model PRO, as presented in Figure 11-8. It should be emphasized that the above-mentioned improvement in the rate of convergence of the RLS algorithm over the LMS algorithm holds only when the measurement error is small compared to the desired response, y_n, that is, when the signal-to-noise ratio is high [47].

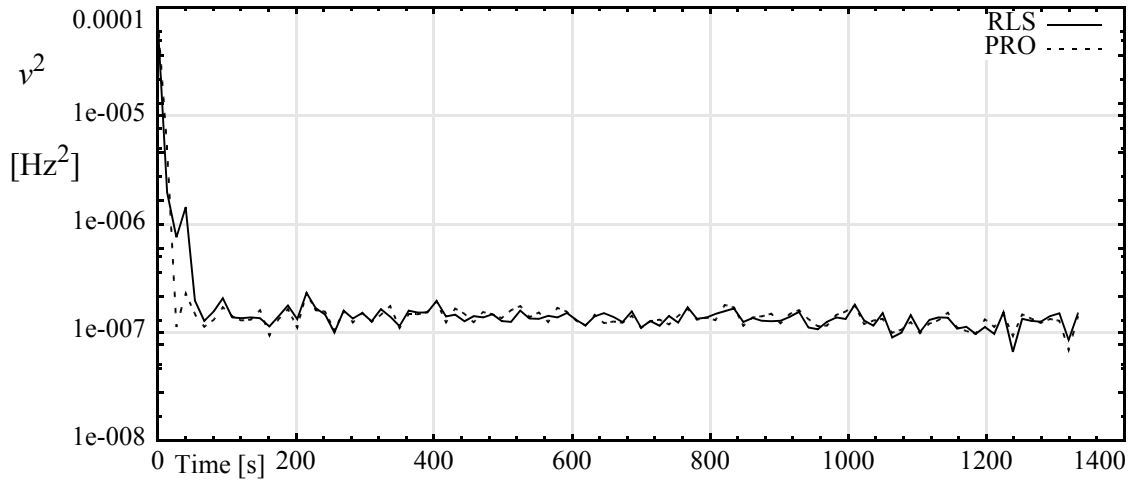


Figure 11-1 Residual comparison between RLS and PRO

LMS algorithm. Convergence of the system identification using the LMS algorithm, in Figure 11-2, is the slowest from all the techniques. This result concurs with that from theory and parametrization. On the other hand, after the *300s* convergence period the performance of the LMS and PRO models are close, which explains why the LMS model is selected by the expert system.

It is interesting to point out that in the very same environment, if the ARMAX form is selected for the power system model,

$$\hat{y}_n = d + \sum_{i=1}^{5} b_i u_{n-i} - \sum_{i=1}^{5} a_i y_{n-i} + \sum_{i=1}^{2 \text{ or } (3\ldots5)} c_i v_{n-i}, \qquad (11.3)$$

the LMS system identification with the same parameter does not converge at all.

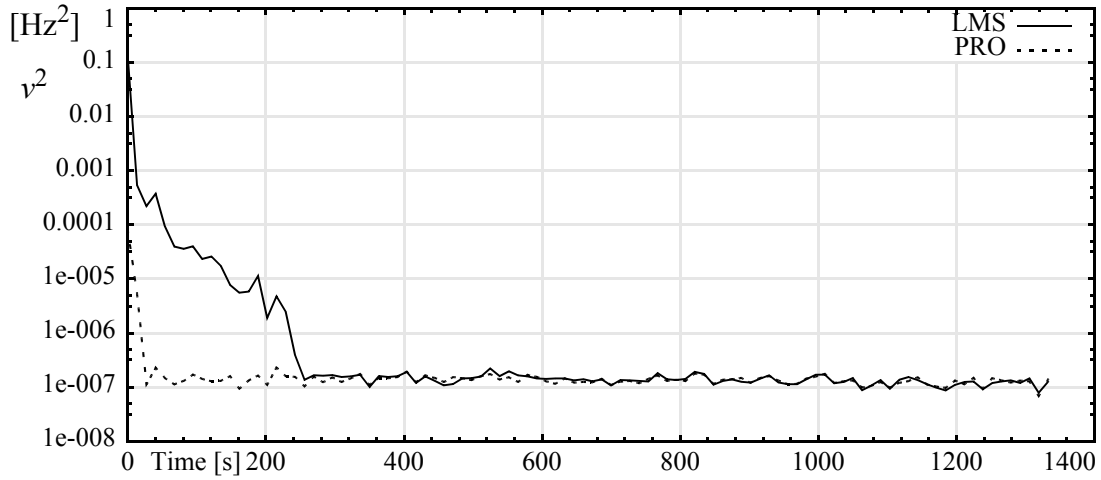


Figure 11-2 Residual comparison between LMS and PRO

WCE algorithm. An example of a non converging system identification is the WCE, shown in Figure 11-3. If the traditional single system identification technique were imple-

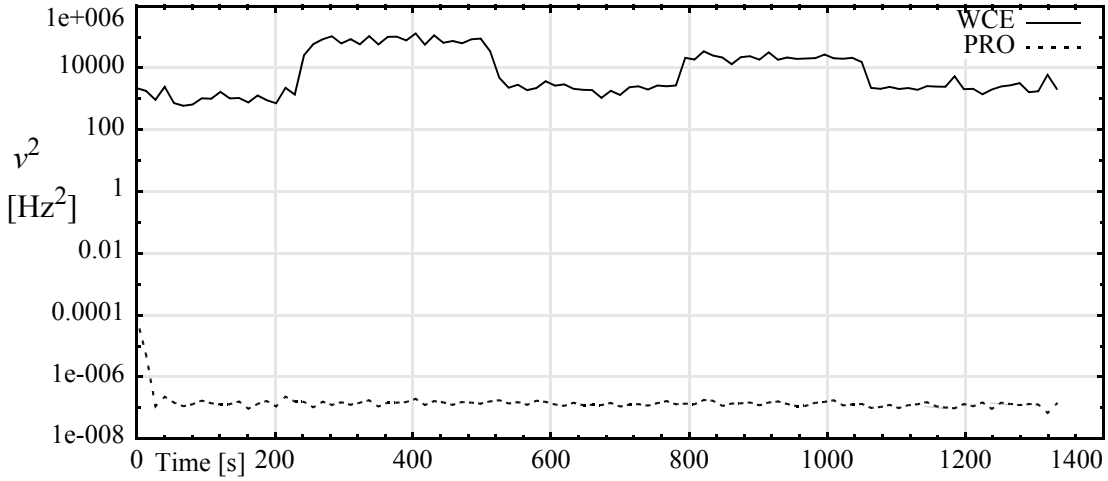


Figure 11-3 Residual comparison between WCE and PRO

mented using the WCE, a disaster in the control loop would result. However, in the multi-

model approach the expert system simply avoids selecting the parameters from the WCE, which eliminates any harm from the poorly behaving algorithm. The five distinguishable regions in the WCE algorithm's chart originate from the five consecutive experiments, which are necessary to collect the 20 minute long measurement sequence.

Kalman Filter. As presented in Figure 11-4, the performance of the system identification using the KF shows a slower convergence with a unsatisfactory initial behavior, represented by a huge error in initial prediction. However, after the convergence of the KF, as expected the resulting system identification is excellent.

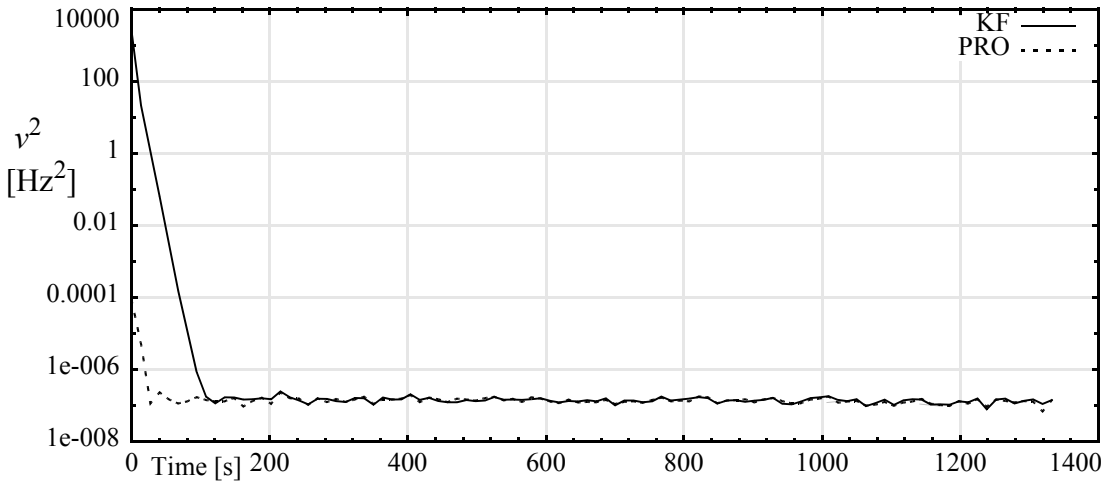


Figure 11-4 Residual Comparison between KF and PRO

IQR algorithm. The initial error of the IQR algorithm is almost as high as that of the KF. Compared to other algorithms, it is followed by the fastest initial convergence, as illustrated in Figure 11-5. Due to this fast convergence the expert system favors the model supplied by the IQR. On the other hand, when the system identification becomes stationary,

the IQR algorithm as it is parametrized, experiences higher volatility in its prediction. Consequently, the expert system prefers other model sources over IQR.

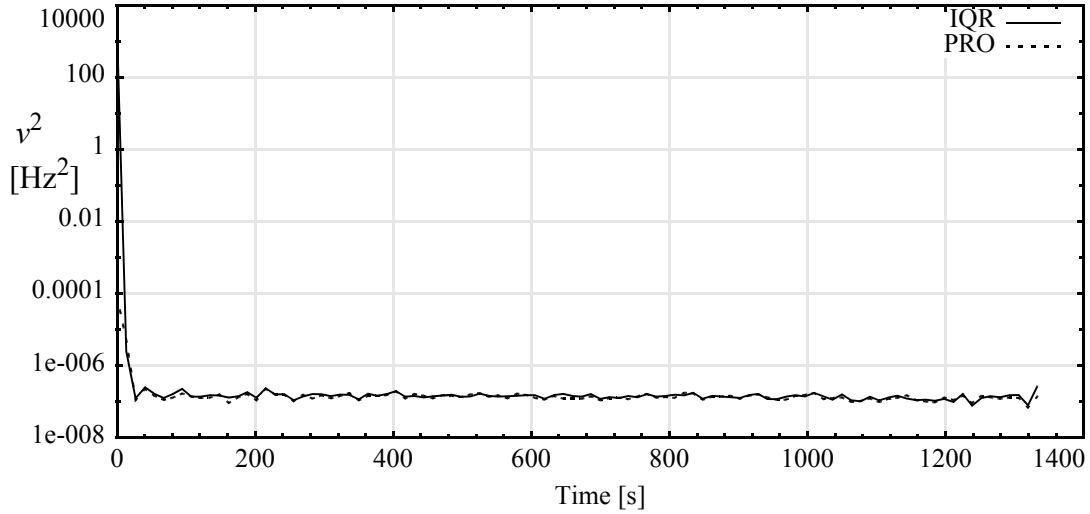


Figure 11-5 Residual comparison between IQR and PRO

FQR algorithm. The FQR algorithm is characterized by a smaller initial prediction error and a very fast convergence, which explains the initial favoritism by the expert system. After the initial period, as seen in Figure 11-6, the residual power of the FQR model is slightly higher than that of the PRO model, which leads to a less favored status of the FQR.

This higher convergence speed can be attributed to a smaller forgetting factor, where both IQR and FQR use $\lambda = 0.975$ while the RLS uses $\lambda = 0.985$. The consequence of this can be observed on the corresponding figures for these algorithms. After the convergence is achieved the algorithm with a smaller forgetting factor or shorter memory demonstrates more volatility in system identification.

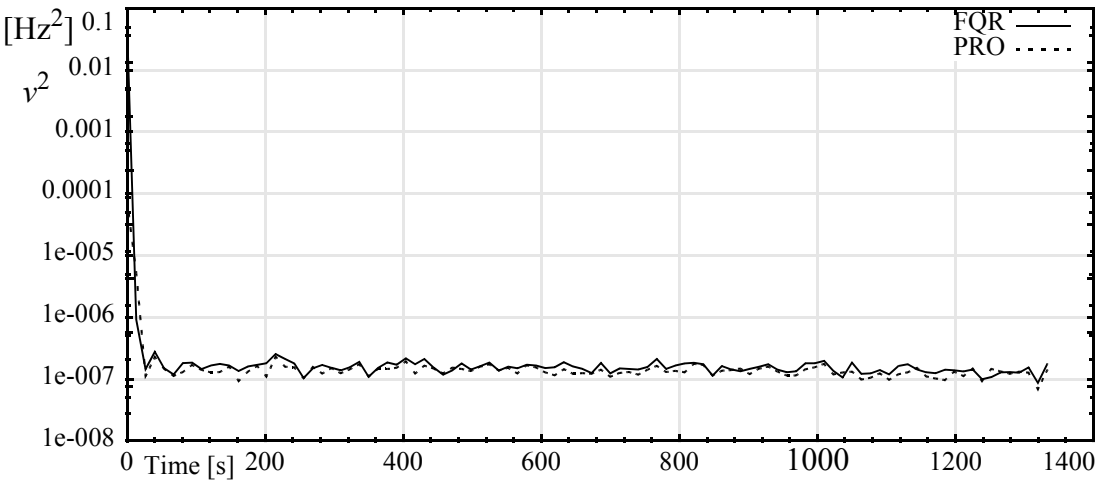


Figure 11-6 Residual comparison between FQR and PRO

QRL algorithm. The QR version of the LMS algorithm, QRL, has a much faster convergence than the LMS as seen from comparing Figure 11-7 and Figure 11-2. Naturally, fast

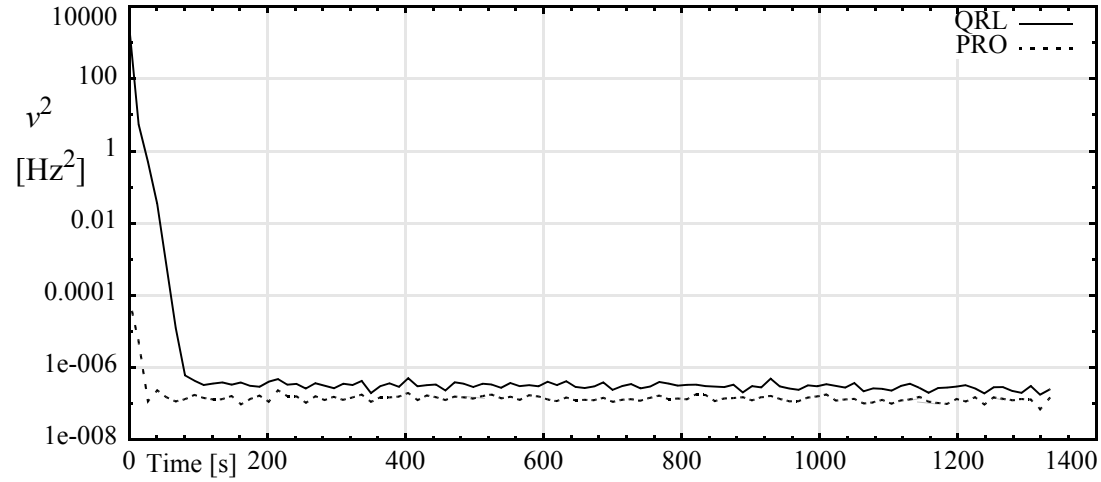


Figure 11-7 Residual comparison between QRL and PRO

convergence leads to system identification with a higher variation in estimated parameters

and higher average error floor. In this particular test, the average prediction error from QRL is always higher than that of the PRO.

11.1.2 Selection by the Expert System

Model selection by the expert system is presented in Figure 11-8. In this figure the ▽ mark shows the source of the model selected by the expert system from the available seven algorithms. During experimental tests, this selection occurs in one second intervals.

Figure 11-8 further supports the results from Figures 11-1 through 11-7. For example,

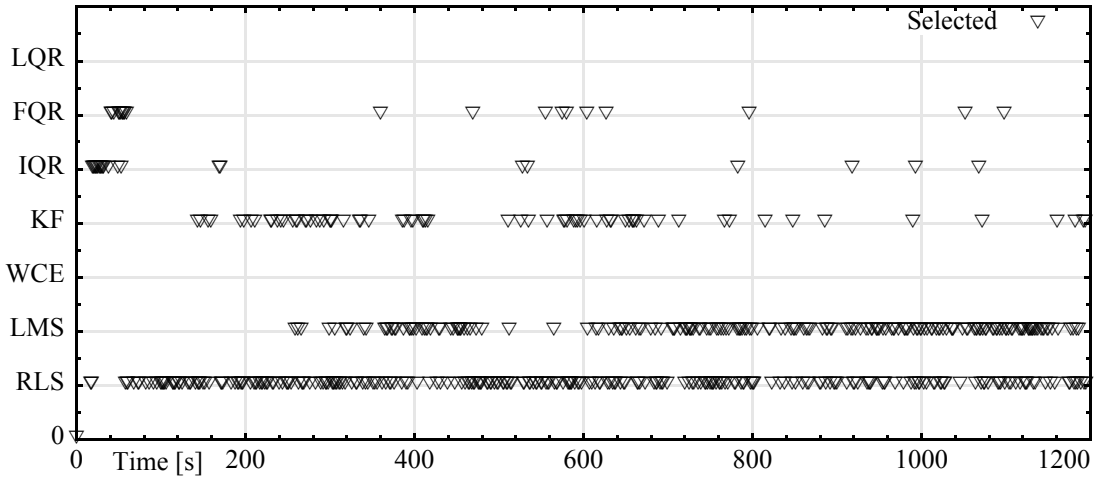


Figure 11-8 Source selection for the PRO

at the start of the system identification process first the IQR algorithm is favored and then the FQR because both are showing fast initial convergence and are delivering the best system model during that time. After the convergence of the RLS, KF and LMS algorithms they are all selected by the expert system. The WCE and QRL algorithms are never selected. The WCE algorithm is clearly unstable while the QRL algorithm delivers a less accurate model than the other system identification techniques.

11.1.3 The Proposed PRO Model

The resulting parameters for the PRO model, selected by the expert system, are presented in Figures 11-9 and 11-10 for the Autoregressive and Moving average parts of the model respectively. The AR parameters demonstrate a stable behavior despite the fact

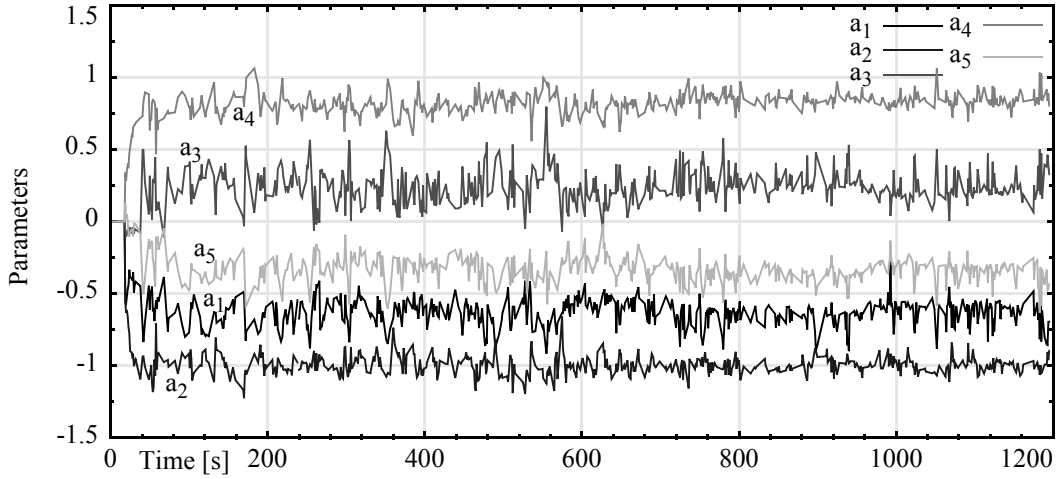


Figure 11-9 The autoregressive parameters of the PRO model

that they are composed from different system identification techniques. The MA parame-

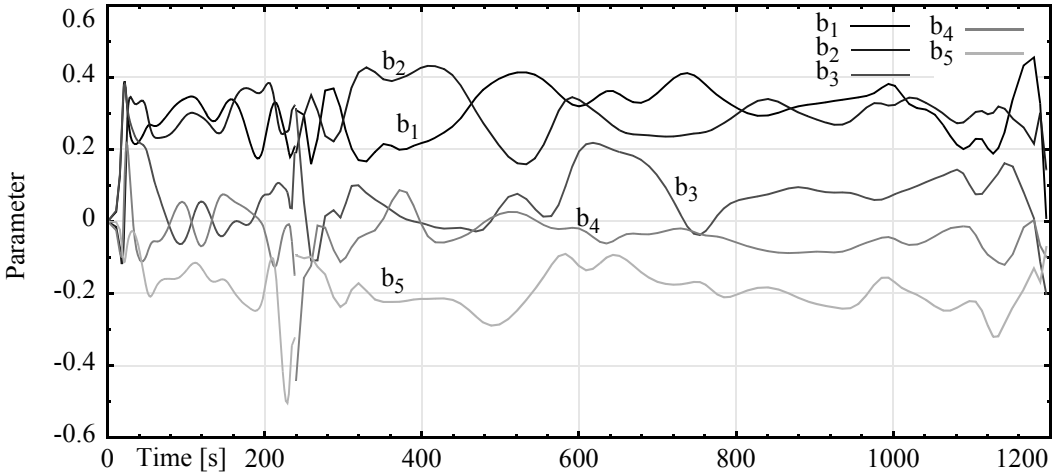


Figure 11-10 The moving average parameters of the PRO model

ters show higher variations, therefore, they can not be observed in their original forms. Thus, they are presented as filtered parameters in Figure 11-10 using the "*smooth bezier*", a built in function of the *Gnuplot.*

11.1.4 Using the PRO Model for Controller Calculation

As the above test results demonstrate, the PRO model is the best representation for the plant to be controlled. Therefore, the PRO model is used in the H_∞ optimization algorithm. The resulting H_∞ MPC, see sections 6.2.2 through 6.2.4, can be considered as a best performing robust controller given the limited a priori knowledge of the plant. The H_∞ MPC is characterized by the achieved infimum, γ, presented in Figure 11-11.

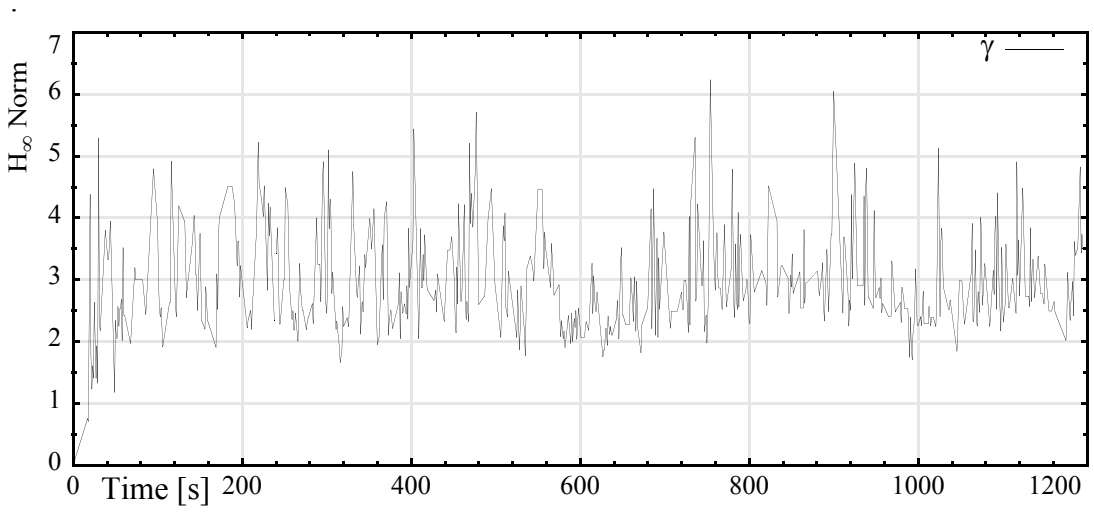


Figure 11-11 The infimum of the H_∞ controller

11.1.5 Conclusion for the System Identification

As a result of the above analysis, the properties of an expert system supervised system identification have been established in a stationary environment. It has also been proved the PRO model is not worse than the models obtained by any single model system identification process. Consequently, the expert system fulfills its tasks.

It is expected that in real implementation, the system will operate in a stationary environment most of the time. However, the control system shall always be ready for occasional perturbations in the system. During the non stationary state of the system, the controller's role is to preserve the system's stability and to steer the process back into its stationary state. As the experiments show, if in a non stationary environment different external disturbances are applied on the power system, the different algorithms behave differently. These tests are presented in sections 11.2 through 11.4 The effects of external disturbances are analyzed through voltage reference change, mechanical input torque change and fault on the transmission line using the possible perturbing methods, as shown in Figure 10-14.

11.2 Voltage Reference Step Change

The voltage reference step change is examined at three different working points of the generator, as described in Experiments 1 through 3.

11.2.1 Experiment 1

In Experiment 1, the micro-synchronous generator is operated at power $P_e = 0.9$ *p.u.* and power-factor $\cos\phi = 0.93$ *lag.* A *2%* step increase in the reference voltage is applied at *1 s* and removed at *5 s*, as presented in Figure 11-12. As a consequence of this perturbation, the generator's power-factor changes to $\cos\phi = 0.91$ *lag* during the *1 s* through *5 s* interval.

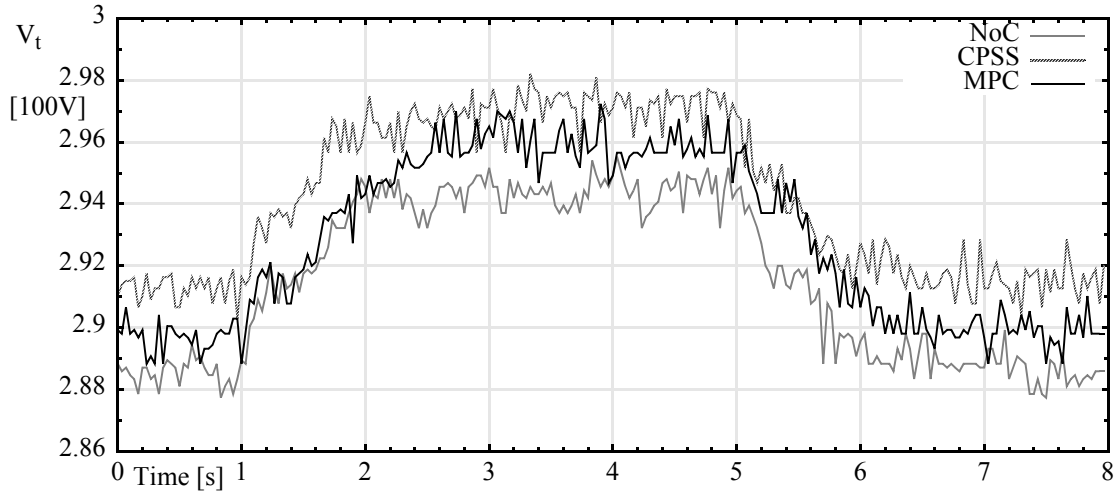


Figure 11-12 Terminal voltage change in Experiment 1

The micro-synchronous generator's rotational speed deviations, $\Delta\omega$, with the proposed H_∞ MPC power system stabilizer, the conventional power system stabilizer (CPSS) and with no controller (NoC) for the *2%* step increase in the reference voltage is shown in Figure 11-13. The amplitude of the oscillations is significantly reduced by the H_∞ MPC.

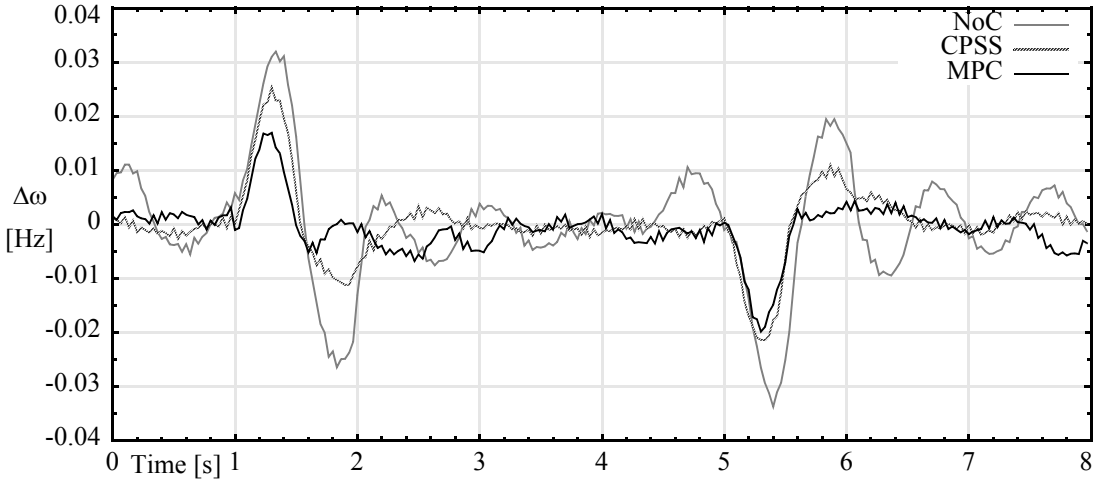


Figure 11-13 Speed deviation in Experiment 1

The control actions of the H_∞ MPC and CPSS are compared in Figure 11-14. The

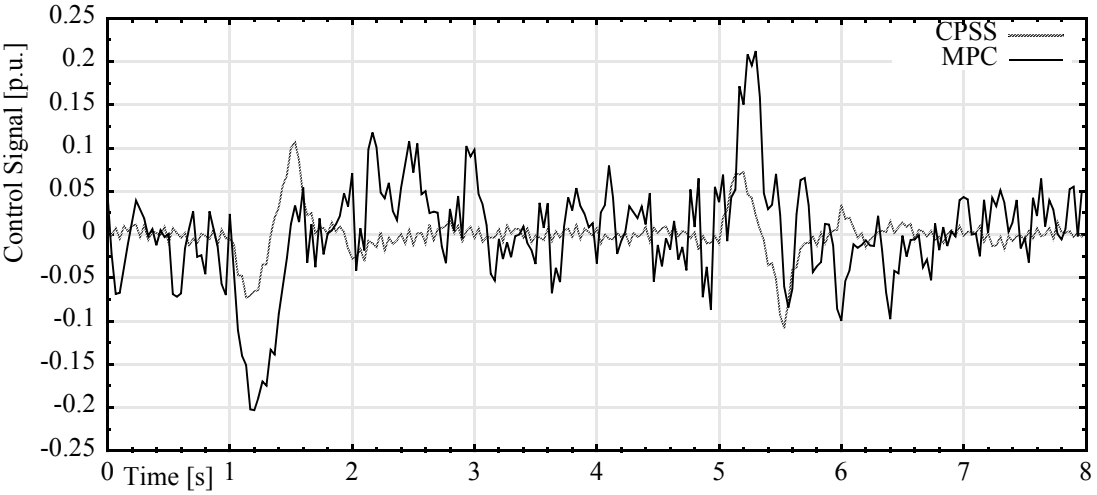


Figure 11-14 Control signal in Experiment 1

fluctuation in the control action of the H_∞ MPC is due to the random signal associated with the system identification algorithm. As described in theory, the system identification requires a random signal for input-output decoupling. The selected signal has a Normal Distribution and it is less than 5% of the control signal. This corresponds to a control

action of 0.05 p.u. Perturbation of the system input causes perturbations in the system output, which is detected by the H_∞ MPC. The controller invokes a response to compensate for these disturbances.

11.2.2 Experiment 2

In this experiment at *2%* step increase in the reference voltage is applied with the electric power operating at $P_e = 0.5$ *p.u.* and $\cos\phi = 0.9$ *lag.* During the higher voltage the power factor is changed to *0.8 lag.* The performances of H_∞ MPC and CPSS are compared as well the speed deviation with NoC is illustrated in Figure 11-15. The system response with H_∞ MPC is significantly better than that with CPSS. While the oscillations with the proposed controller last only for 0.5 s, the oscillations with the conventional controller diminish after 1 s.

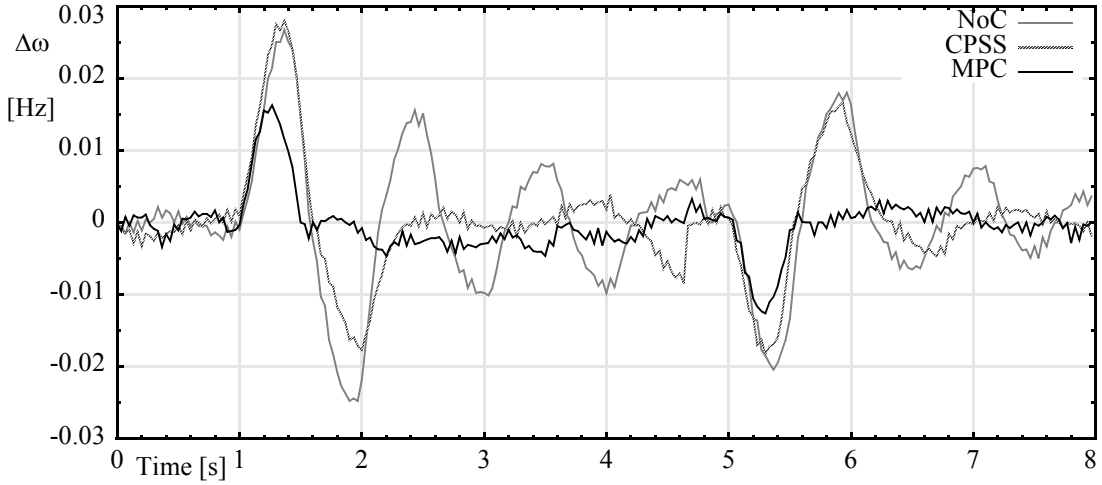


Figure 11-15 Speed deviation in Experiment 2

Comparison of the control actions of the H_∞ MPC and CPSS is provided in Figure 11-16.

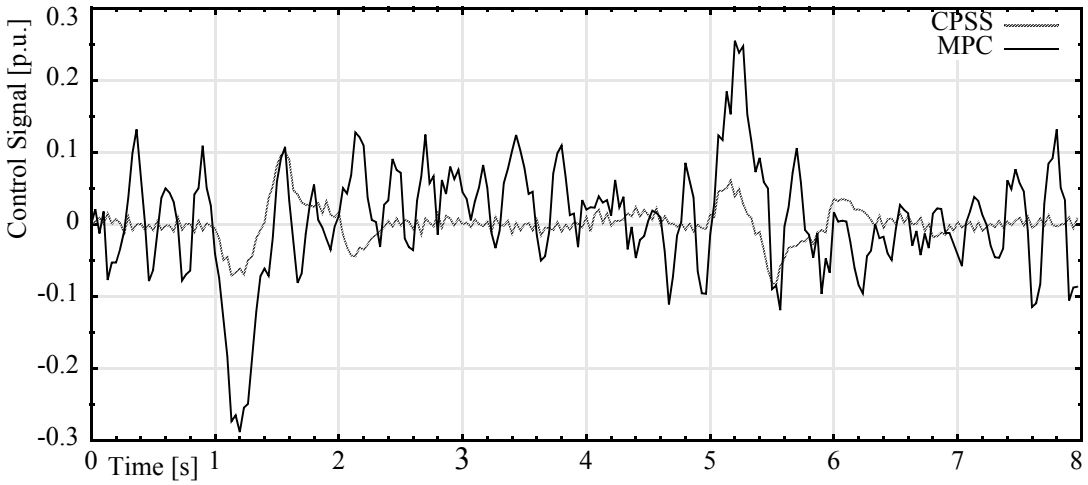


Figure 11-16 Control action in Experiment 2

11.2.3 Experiment 3

The voltage reference step change of 2% is also tested with a leading power factor when the generator operates with the electric power $P_e = 0.5\ p.u.$ and power-factor $\cos\phi =$ *0.9 lead.* During this experiment the power-factor changes to $\cos\phi = 0.8$ *lead.* The micro-synchronous generator's rotational speed deviation, $\Delta\omega$, with the H_∞ MPC, CPSS and NoC is demonstrated in Figure 11-17. Compared to the amplitude of oscillations with CPSS, the amplitude of the oscillations with H_∞ MPC is reduced almost to half.

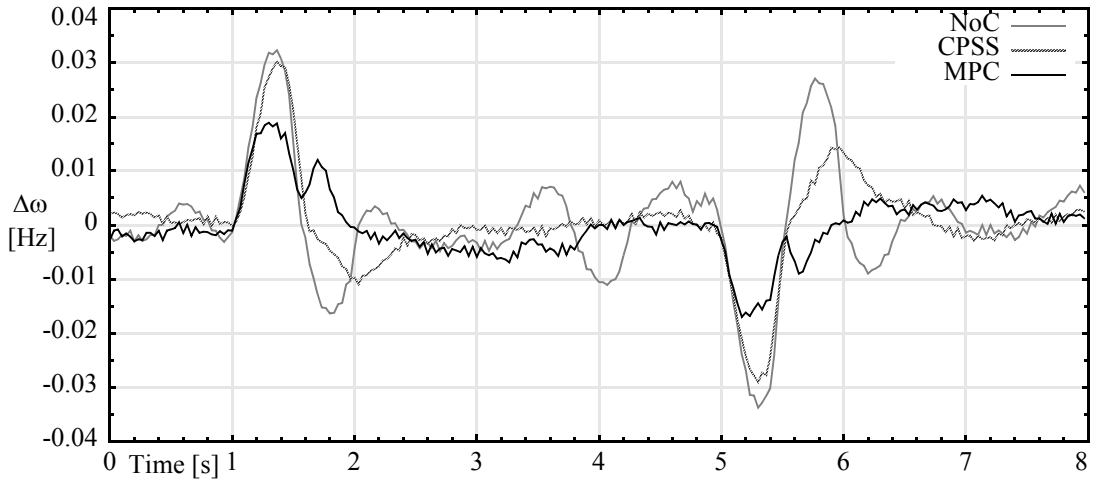


Figure 11-17 Speed deviation in Experiment 3

The control actions of the H_∞ MPC and CPSS are compared in Figure 11-18.

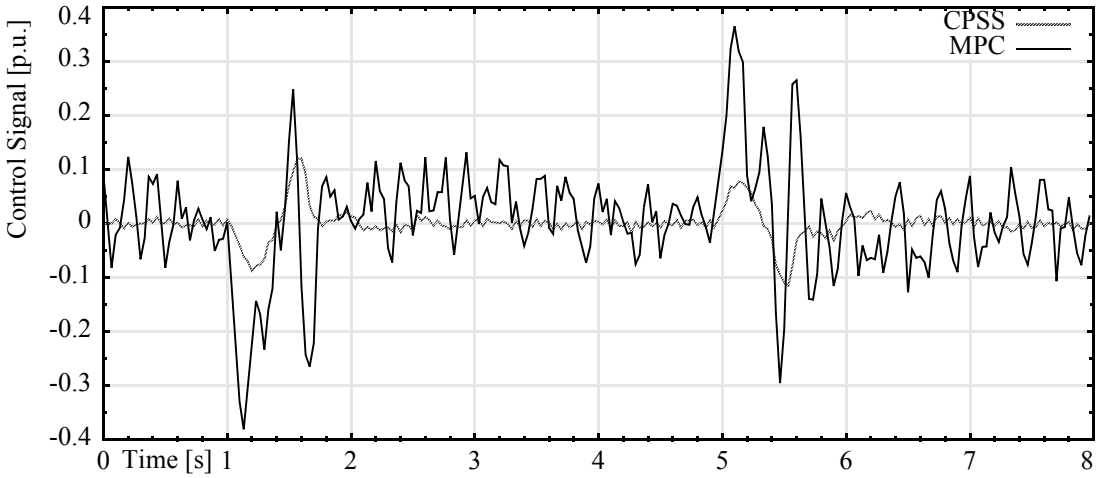


Figure 11-18 Control action in Experiment 3

The above presented test results verify that both control algorithms perform well at all operating points for this type of disturbance. The difference in performance goes in favor of H_∞ MPC, which has smaller amplitudes in post-disturbance oscillations.

11.3 Input Torque Step Change

The input torque step change disturbance is examined at three different working points of the generator, as described in Experiments 4 through 6.

11.3.1 Experiment 4

In this experiment the micro-synchronous generator is operated at power $P_e = 0.9\ p.u.$ and power-factor $\cos\phi = 0.93\ lag$. A *18%* step decrease in the input torque is applied at *1 s* and removed at *5 s*. During this time the system operating point changes to a power of $P_e = 0.725\ p.u.$ and power-factor $\cos\phi = 0.91\ lag$. This step change in the electric power of the micro-synchronous generator with the H_∞ MPC, CPSS and NoC is shown in Figure 11-20.

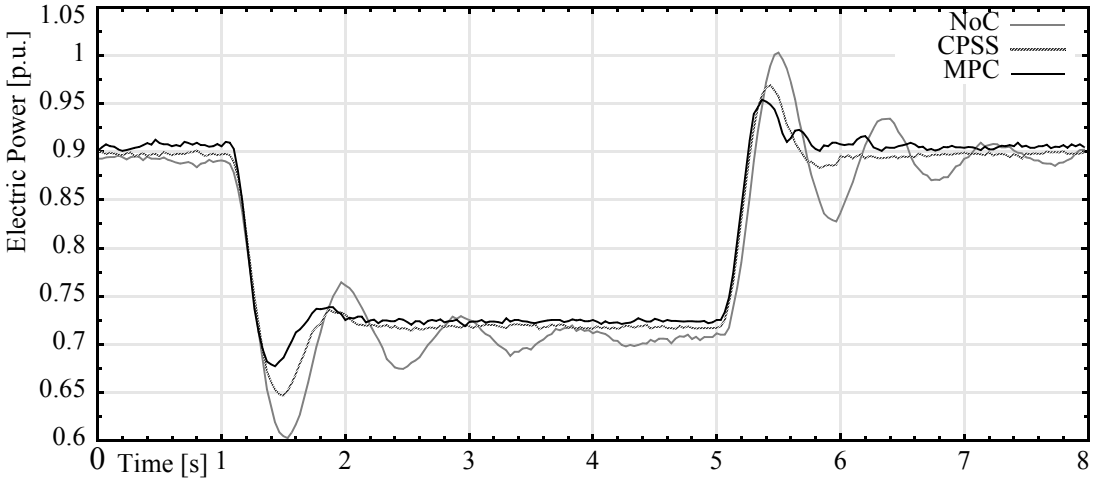


Figure 11-19 Electric power in Experiment 4

The micro-synchronous generator's rotational speed deviation, $\Delta\omega$, with the H_∞ MPC, CPSS and NoC is shown in Figure 11-20. In this experimental situation the H_∞ MPC provides better reduction in the amplitude of the oscillations but the CPSS shows faster settling time of the response after the second disturbance. This can be explained as a result of H_∞ controller action, whose primary objective is to reduce the effects of larger disturbances and ignore the smaller ones.

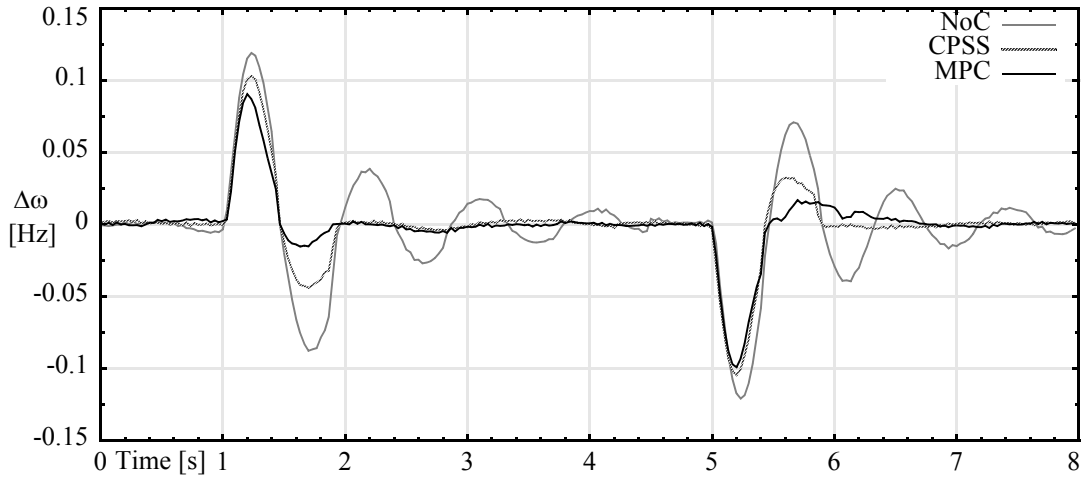


Figure 11-20 Speed deviation in Experiment 4

The investigation of control actions with H_∞ MPC and CPSS is presented in Figure 11-21.

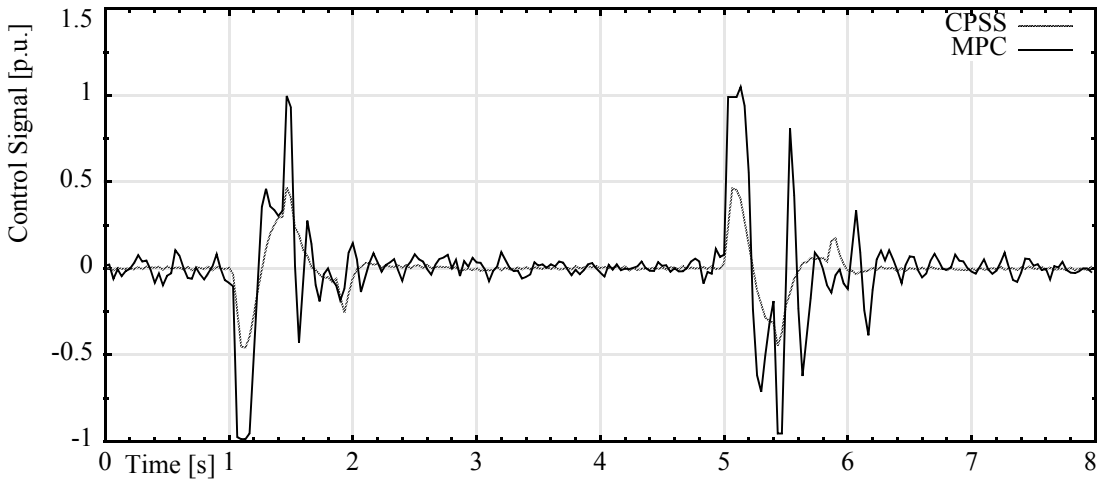


Figure 11-21 Control action in Experiment 4

To examine the system identification during this major disturbance, the residuals from the PRO model and other system identification techniques are plotted in Figure 11-22. This figure shows that the residual power from the PRO model usually stays below all the others, which justifies the expert system selection.

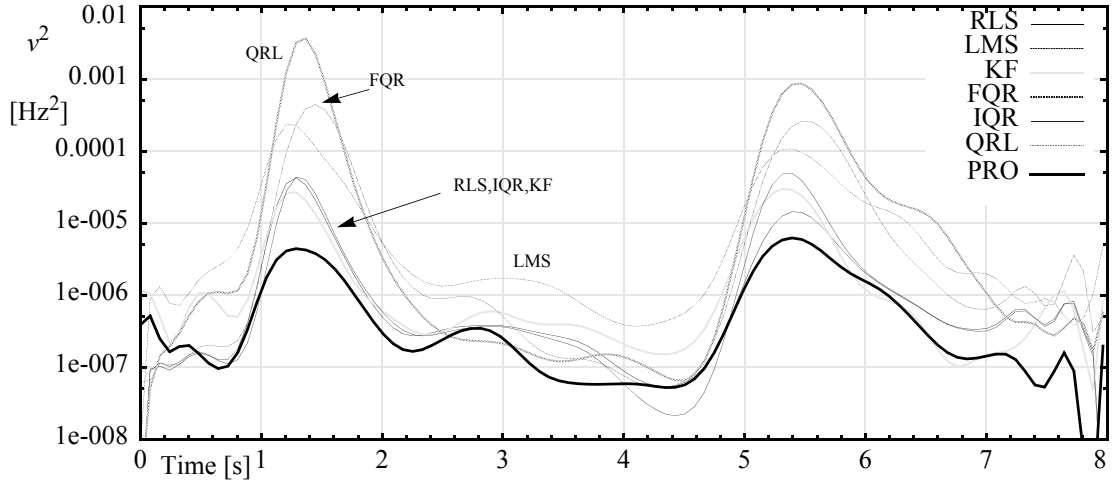


Figure 11-22 System identification during Experiment 4

11.3.2 Experiment 5

The input torque step change experiment is performed with power $P_e = 0.5$ *p.u.* and $\cos\phi = 0.9$ *lag.* When a *0.1 p.u.* step change is introduced during the time interval of *1 s* through *5 s*, the operating point changes to $P_e = 0.4$ *p.u.* and $\cos\phi = 0.8$ *lag.* System responses with the H_∞ MPC, CPSS and NoC are illustrated in Figure 11-23. While the oscillations for both controllers last for 2 s and 2.5 s, the amplitude of the oscillations are much faster reduced with H_∞ MPC and that brings the system closer to its stable state.

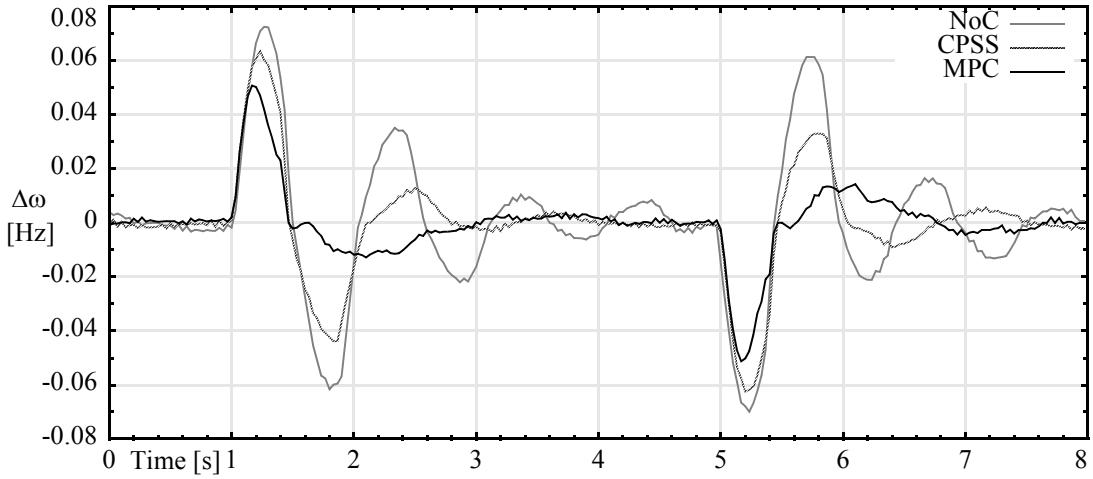


Figure 11-23 Speed deviation in Experiment 5

The control signals for H_∞ MPC and CPSS are shown in Figure 11-24.

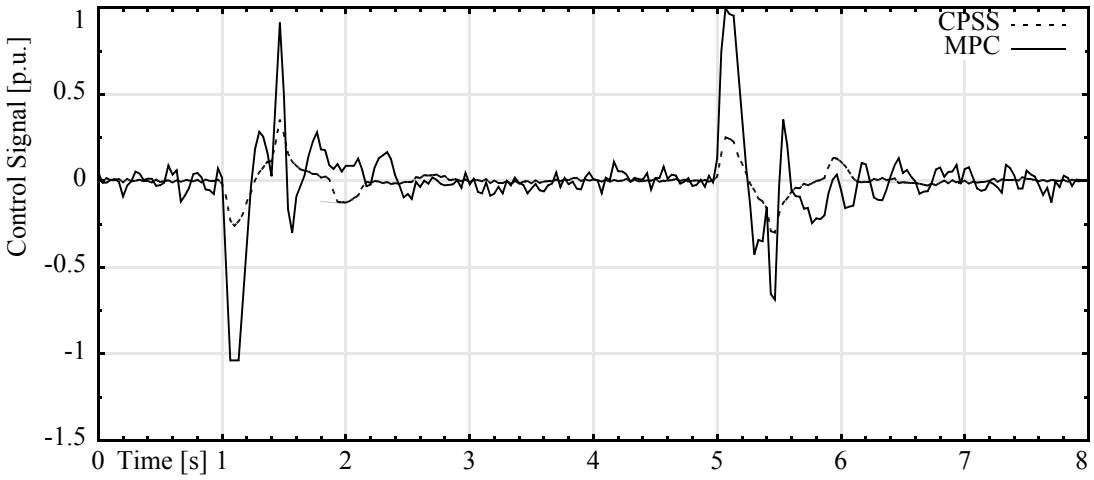


Figure 11-24 Control action in Experiment 5

11.3.3 Experiment 6

A *0.1 p.u.* step decrease in the input torque is also tested with a leading power factor. During this experiment, the generator's electric power $P_e = 0.5$ *p.u.* with $\cos\phi = 0.9$ *lead*, changes to $P_e = 0.4$ *p.u.* with $\cos\phi = 0.93$ *lead.* Speed deviations with H_∞ MPC, CPSS and NoC are presented in Figure 11-25. At the above indicated operating point the system response with the H_∞ MPC shows superior performance compared to that of the CPSS. The oscillations with the H_∞ MPC last for 0.5 s while the oscillations with the CPSS last for 1 s. At the same time the maximum amplitude of oscillations is also reduced with the H_∞ MPC.

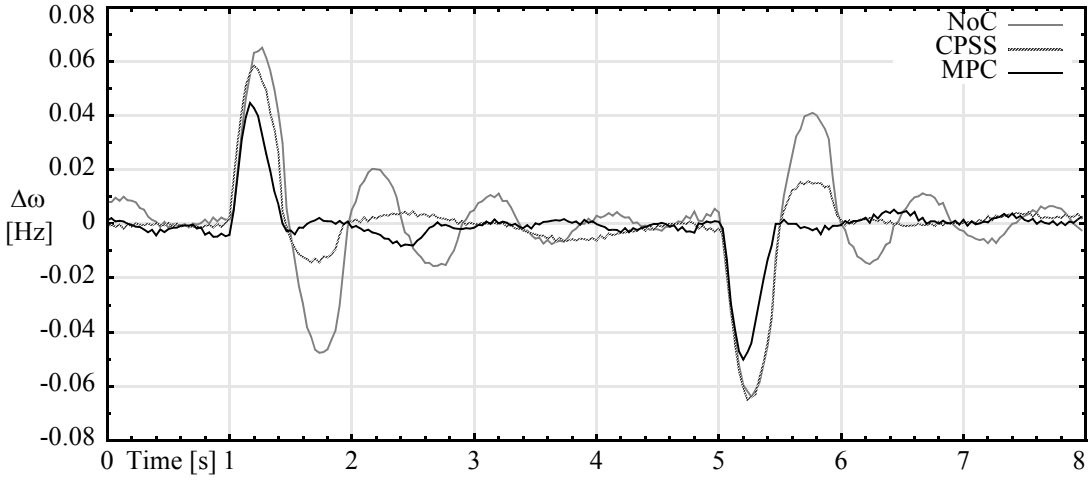


Figure 11-25 Speed deviation in Experiment 6

The control actions of H_∞ MPC and CPSS are examined in Figure 11-26.

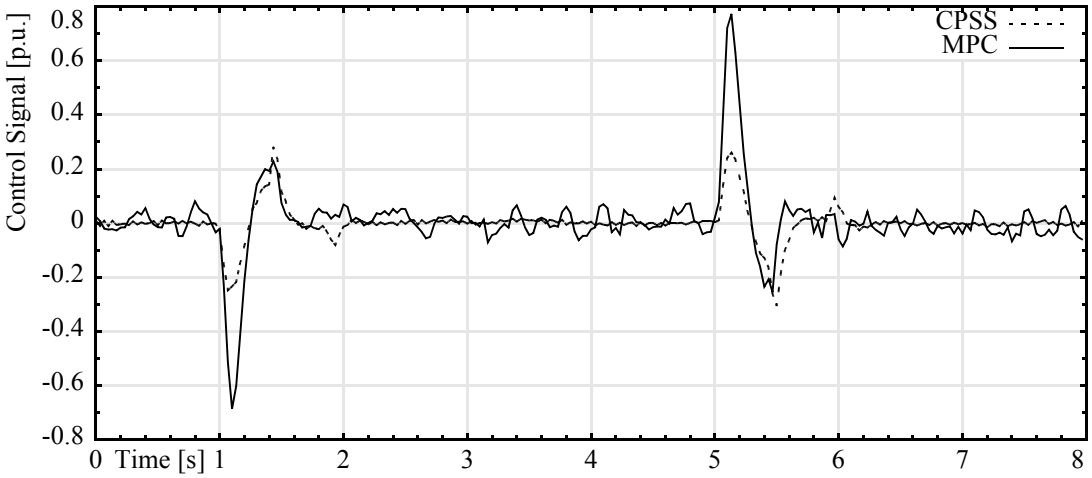


Figure 11-26 Control action in Experiment 6

11.4 Three-Phase to Ground Fault Test

The three-phase to ground fault is examined at three different working points of the generator, as described in Experiments 7 through 9.

11.4.1 Experiment 7

To investigate the performance of the H_∞ MPC under transient conditions caused by transmission line fault, a three phase to ground fault test has been conducted. In Experiment 7, the micro-synchronous generator is operated at power $P_e = 0.9\ p.u.$ with power factor $\cos\phi = 0.93\ lag.$ Under this operating condition, with both lines in operation, a three phase to ground fault is applied in the middle of one transmission line at *1 s* The faulty transmission line is opened by relay action at both ends *100 ms* later. The first unsuccessful closure attempt is made after *600 ms,* and the line is re-opened again after *100 ms* due to a permanent fault. During the fault state the micro-synchronous generator is operated at power $P_e = 0.9\ p.u.$ with $\cos\phi = 0.85\ lag.$ The second successful closure attempt is applied at *8 s* and the system returns to its initial operating condition. This test is best characterized by a change in the generator's terminal voltage, V_t. Prior to the fault the energy is transported through two transmission lines. Following the fault the same energy must be transported through only one transmission line, which explains the terminal voltage increase during the fault, as shown in Figure 11-27.

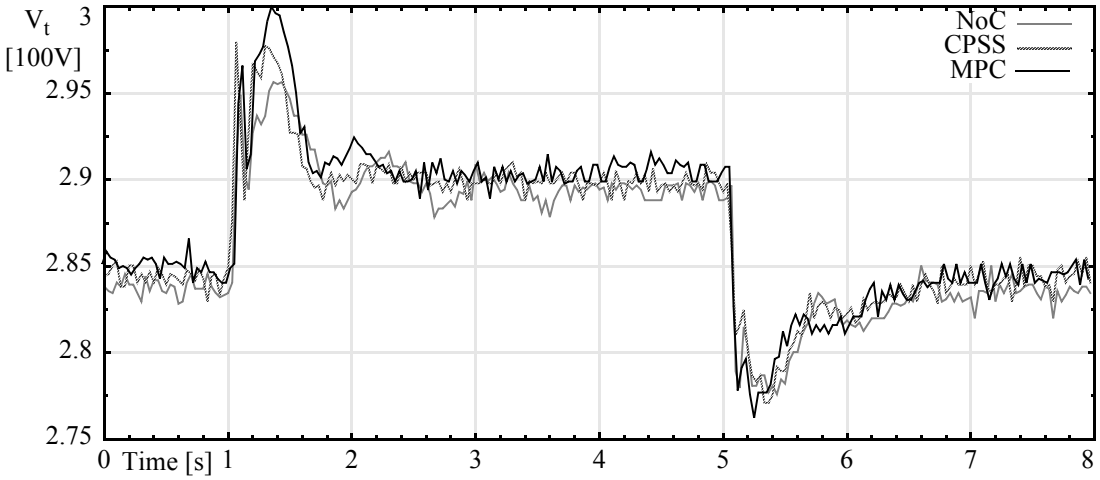


Figure 11-27 Terminal voltage change during Experiment 7

Under the above transient conditions, system responses with the H_∞ MPC and CPSS are compared in Figure 11-28. It can be observed that the H_∞ MPC outperforms the CPSS with smaller overshoot and faster settling time in both cases, at 1 s and 5 s

.

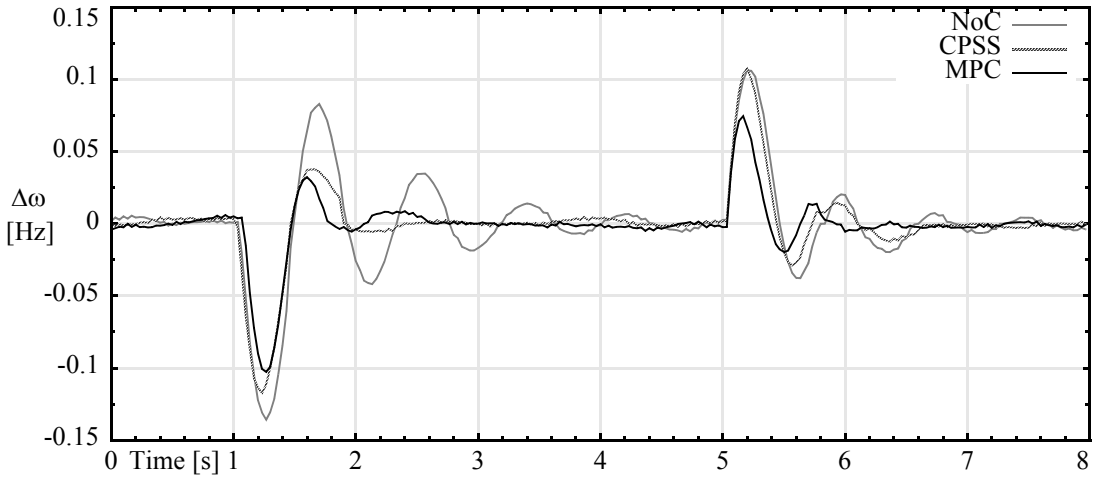


Figure 11-28 Speed deviation in Experiment 7

Control actions of H_∞ MPC and CPSS under transient conditions are also investigated. Figure 11-29 outlines the test results.

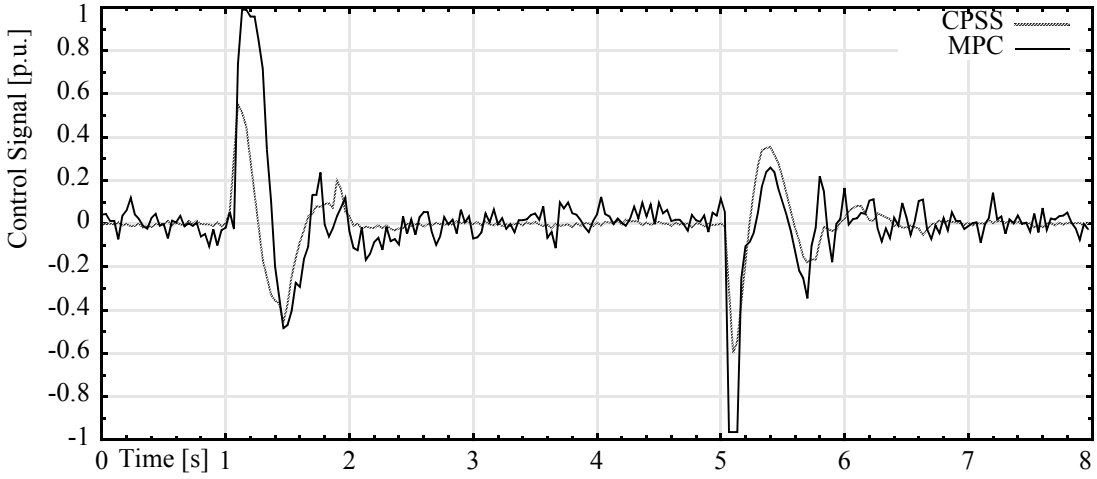


Figure 11-29 Control action in Experiment 7

To examine the system identification during this power line fault, the residuals from the PRO model along with other system identification techniques are plotted in Figure 11-30. This graph shows that the residual from the PRO model usually stays below all the others, by which the expert system selection is justified. The first and last 0.5 s on the Fig-

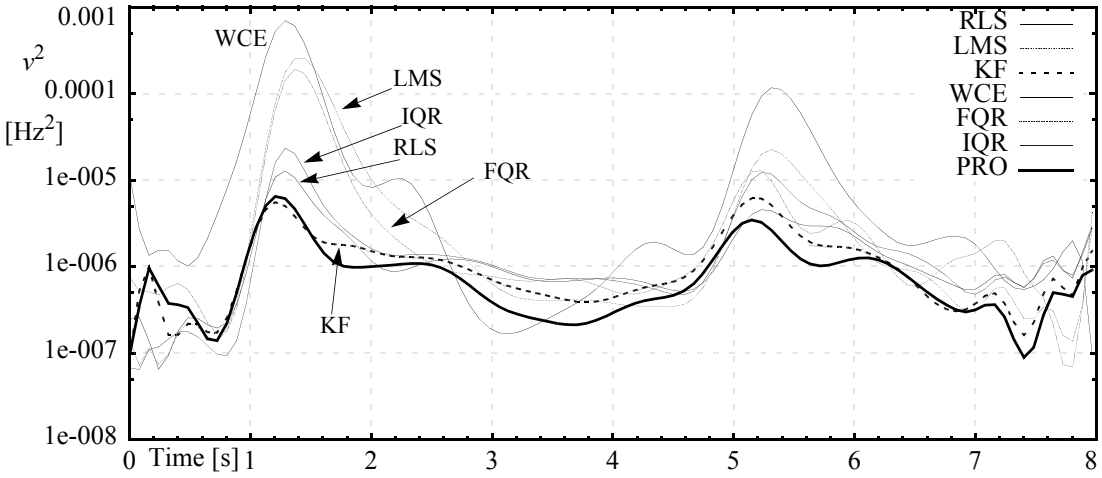


Figure 11-30 System identification in Experiment 7

ure 11-30 is a distorted representation of the measurements which is caused by the smoothing function "*smooth bezier*".

11.4.2 Experiment 8

Responses to a three phase to ground fault disturbance are compared in Experiment 8, where the micro synchronous generator's power is $P_e = 0.5\ p.u.$ with $\cos\phi = 0.9\ lag.$ As seen in Figure 11-31, during the fault the micro-synchronous generator is operating at a power of $P_e = 0.5\ p.u.$ with a power factor $\cos\phi = 0.8\ lag.$ Although the well tuned CPSS reduces the oscillations, the H_∞ MPC exhibits excellent damping properties.

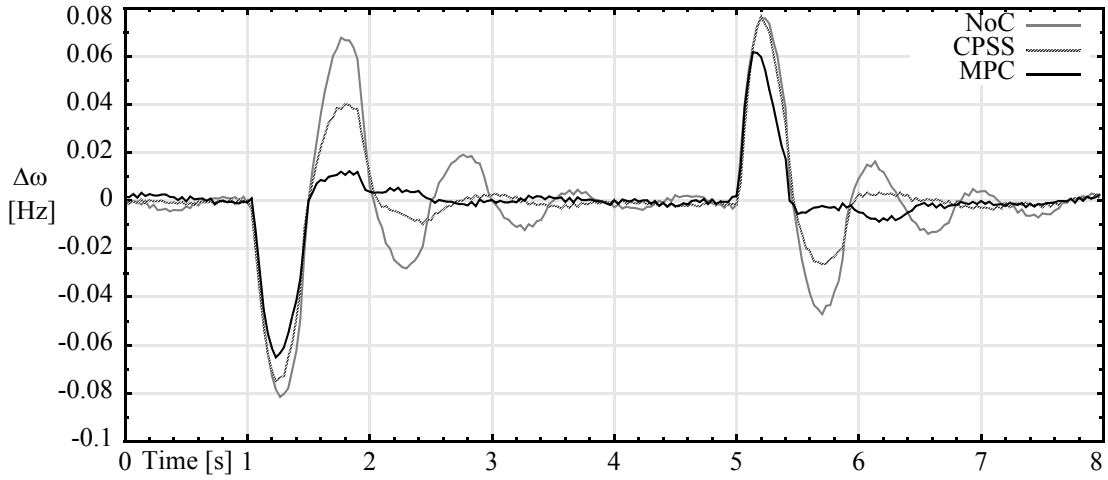


Figure 11-31 Speed deviation in Experiment 8

The control actions of H_∞ MPC and CPSS are examined in Figure 11-32.

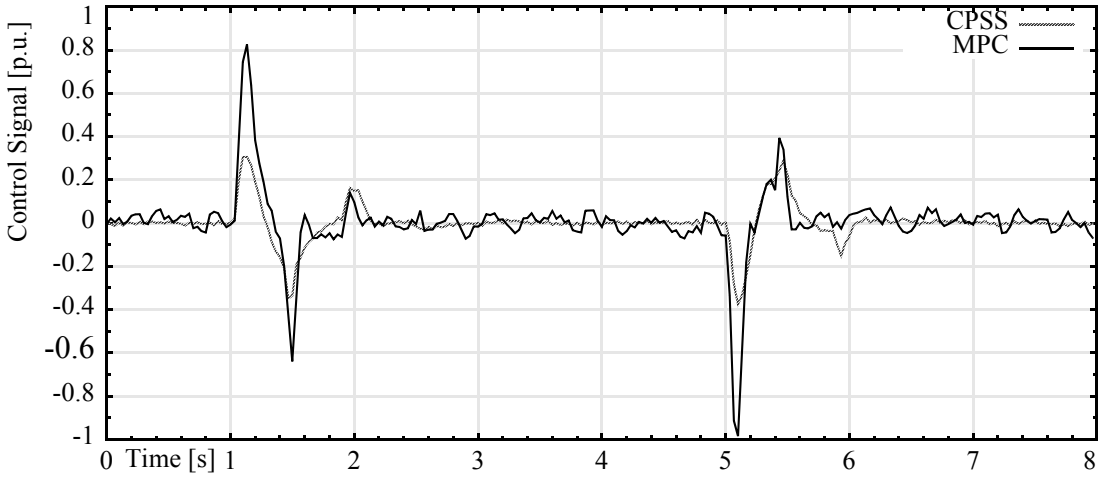


Figure 11-32 Control action in Experiment 8

11.4.3 Experiment 9

Performance in response to another three phase fault is presented in Figure 11-33. The micro synchronous generator's operating power is $P_e = 0.5\ p.u.$ with a power factor

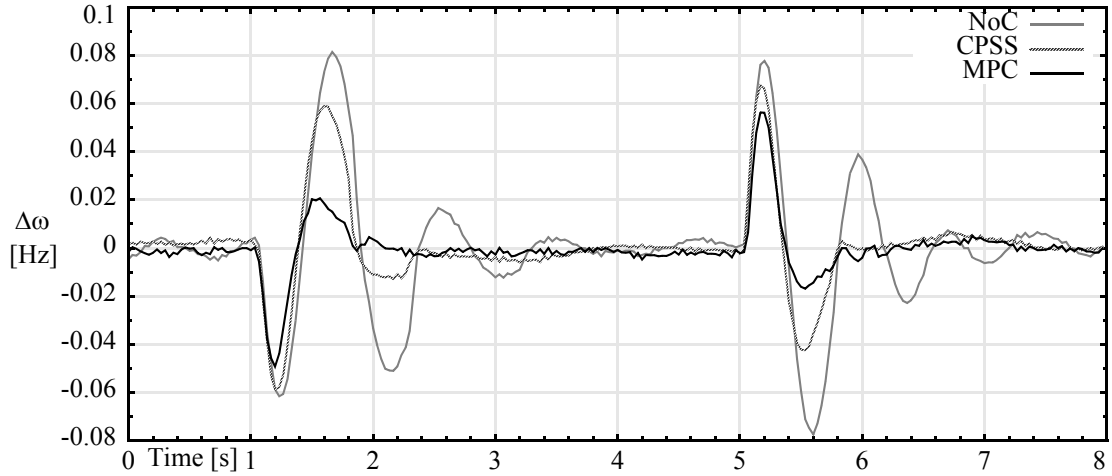


Figure 11-33 Speed deviation in Experiment 9

$\cos\phi = 0.9$ *lead.* During the fault, the micro-synchronous generator operates at a power $P_e = 0.5$ *p.u.* with a power factor $\cos\phi = 0.8$ *lead.* It is obvious that the H_∞ MPC provides an effective way to improve the generator's stability under these operating conditions, where a very fast control action brings the amplitude of the oscillations down.

In Figure 11-34 the control signals for H_∞ MPC and CPSS are compared.

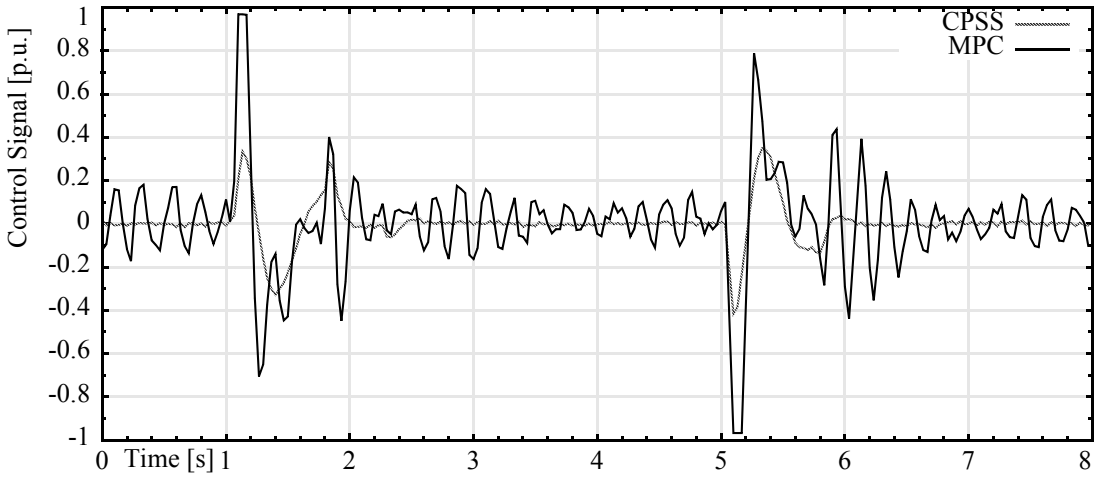


Figure 11-34 Control action in Experiment 9

11.5 Summary

In a real-time digital control environment, described in Chapter 10, an H_∞ MPC power system stabilizer, discussed in Section I and II, has been implemented and tested on a physical model of a power system. The experimental results obtained with the H_∞ MPC are compared with that of a digitized conventional power system stabilizer CPSS, described in section 10.3.2. The test results, presented on graphs, have proved that both

the CPSS and the proposed H_∞ MPC are very effective in improving the dynamic performance of the model power system. Furthermore, both control methods keep their abilities for any operating point of the system. Yet, a clear difference in the performance of these two control methods can be observed. This difference originates in the derivation of these methods. The CPSS is based on a conventional controller design using proportional-integral compensator with lead-lag elements, which are fine-tuned around one operating point. The H_∞ MPC is designed using optimization techniques with the objective to reduce the effects of disturbances with highest magnitude. The test results clearly show that the system outputs with larger deviations are much more reduced with the H_∞ MPC than with the CPSS.

This difference in reducing effects of disturbances with highest magnitude originates from the weaker control action of the CPSS. Figures that illustrate control actions from the above experiments clearly attest to this argument. To verify that with a higher CPSS gain $K_S = -0.4$, see Table 10-3, the weakness in the performance of the CPSS can be improved, one last experiment was conducted.

11.5.1 Experiment 10

In Experiment 10, the generator's operating power is Pe = *0.5 p.u.* with cosϕ=*0.93 lag.* At time *1 s*, a *0.12 p.u.* step decrease in torque reference is applied to the system. The operating power of the generator returns to its original value at time *5 s*. As Figure 11-36 illustrates, when the CPSS gain is $K_S = -0.4$ the control action of the CPSS is improved.

However, the resulting system stability is compromised by longer lasting oscillations, as shown in Figure 11-35. Overall, performance of the CPSS deteriorates when the operating

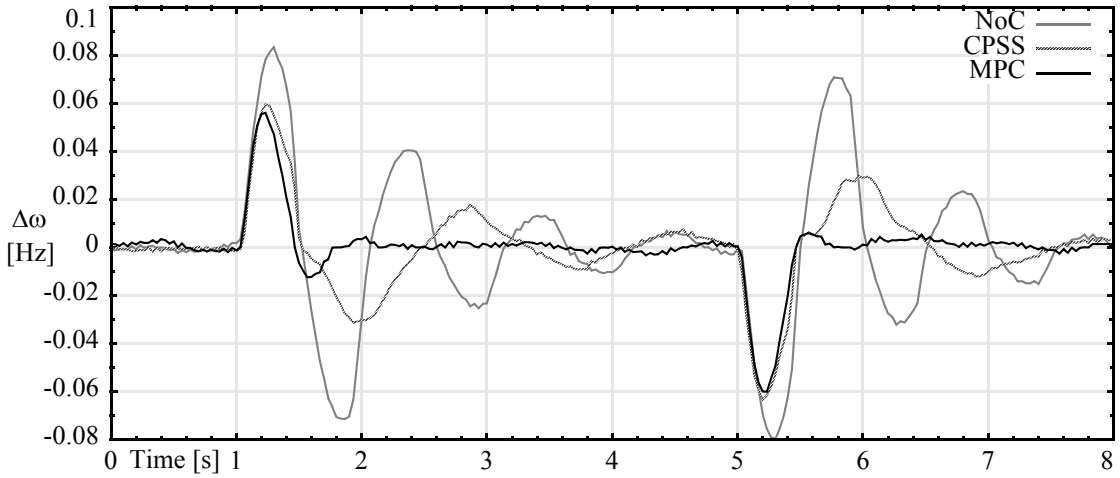


Figure 11-35 Speed deviation in Experiment 10

conditions deviate from the design point. These experiments also prove that the H_∞ MPC

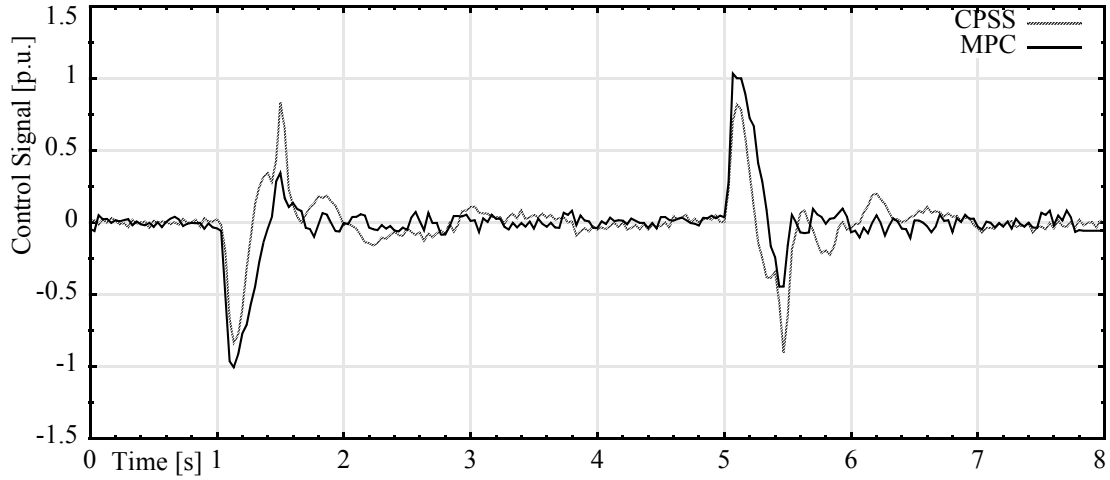


Figure 11-36 Control action in Experiment 10

is able to adapt itself to new operating conditions and it can produce consistently good performance in accordance with the optimization criterion. These experimental results are consistent with the theoretical and simulation results and demonstrate that the proposed H_∞ MPC stabilizer outperforms the CPSS.

12 Conclusions and Recommendation

12.1 Conclusions

One can reasonably expect that future industrial systems will become more complex. Consequently, improved performance requirements will be established for industrial controllers. Therefore, conventional three-term-controllers may not be able to resolve the need for higher accuracy and robustness in these industrial environments. The next generation of controllers will be expected to handle the working point changes of a nonlinear plant and they should perform well under system disturbances. It is also expected that the control systems of the future will require minimal studies prior to their implementation.

A controller that could handle the above requirements is an adaptive controller with the control action that is optimal for both system identifications and implementation of the control. The focal point of this thesis is to develop a controller that will meet the above expectations. The controller design is based on the estimation of the current plant model and the information obtained from the estimate is used to redesign the controller. This thesis proposes a two step adaptive solution for a robust control in the form of an H_∞ MPC algorithm, where considerable attention is directed towards the reduction of prior studies.

In step one, the H_∞ MPC algorithm performs a robust system identification using a general method for estimating discrete unknown parameters associated with robust control uncertainty models. The key element of this robust system identification is an expert sys-

tem, which supervises multiple concurrently performing system identification techniques and periodically selects from the estimated parameters to define the optimal system model. This concept requires that the concurrently performing system identification techniques have distinct estimation properties. The difference in behavior is accomplished either by using different algorithms, such as the RLS, LMS, WKE, KF, FQR, IQR, and QRL or by parametrizing one algorithm with λ_{MIN}, λ_{MED}, and λ_{MAX} parameters.

The different behavior of system identification techniques is used with the assumption that they will behave differently under different perturbations and thereby, at least one will provide a chance for the expert system to select a proper model. In order to obtain a system model which is relevant to the plant control, the system identification is performed in a closed-loop. As test results show, with this approach the robustness of system identification is significantly improved.

In step two, the H_∞ MPC algorithm uses the proposed system model from step one to solve the corresponding robust control problem by the H_∞ (sub)optimization technique. A detailed implementation procedure for this control algorithm, which is obtained by the game theory approach, is provided in this thesis. The H_∞ controller is formulated using the solutions of two recursive Riccati equations, one for the controller and one for the state estimator. The acquired H_∞ design technique is one of the most advanced methods available today for designing robust controllers. One big advantage of this technique is that it allows the designer to tackle the most general form of control architecture, while explicitly considering such factors as uncertainties, disturbances, actuator/sensor noises, and performance measures. This is accomplished through four weighting matrices. In the proposed

design the selection of weighting matrices is simplified by a coefficient that multiplies a unity matrix. The weighting matrices with the system model define the infimum, γ, of the H_∞ controller. If the value of γ is smaller, the controller is more robust with a conservative control action. However, if γ increases, the control action approaches to that of LQR. It is a design trade-off to select those weighting matrices that deliver a desired control action.

During the simulation study and in the experimental tests, the H_∞ MPC was applied as a power system stabilizer and tested under various conditions. For comparison, a digitized conventional power system stabilizer, CPSS, has also been implemented and tested under the same conditions. The H_∞ MPC has proven to be wide ranging in action and appropriate for steady-state, dynamic and transient operating conditions. It has provided good post-fault recovery for the tested disturbances, such as the reference voltage step change, input torque step change and three phase to ground fault. These results are consistent with the expectations, demonstrating the correctness and superiority of the proposed H_∞ MPC over the CPSS results.

The simulation and experimental results have also confirmed that the introduced expert system increased the effectiveness and robustness of the real-time system identification techniques.

The implementation of concurrently running algorithms has been greatly simplified by the C++ object oriented programming language. This programing language not only simplifies the multiple usage of a function like the system identification technique but it also enables the development of common library functions which can be used in different algo-

rithms ensuring that all the algorithms use the same functions while the parameters in each algorithm are defined in a unique way.

The outcome of this thesis is a systematic, ready for application methodology for building an H_∞ MPC algorithm. This has been accomplished by translating the complex mathematical theory of control into a format that is easy to comprehend. This research also contributes to a thorough understanding of the application of H_∞ MPC to industrial system control problems and it demonstrates how the H_∞ MPC can be successfully applied in an industrial design.

12.2 Recommendation for Future Research

The idea behind developing H_∞ MPC was to introduce a controller which requires a minimum of preliminary research and tuning for a particular implementation. Several important findings have emerged as a result of this research. However, there are still opportunities for further improvements and development. In the following paragraphs a few feasible and required improvements are proposed.

- **Preliminary controller** - During the initial phase of the H_∞ MPC implementation, the controller is not defined for a period of time. Taking into account that during this phase some systems may be inherently unstable, the need for an integrated preliminary simple controller is necessary. Starting with an initial feedback controller, an iterative scheme of modelling and designing the MPC will provide information on the dynamics of the plant relevant to feedback system identification and the consequent robust control.

- **Model order selection** - Estimating the appropriate model order of a dynamic system from the observed input-output data is an important procedure in adaptive control. Neither having a more complex model nor having an overly simplified one has advantages. The literature on model order selection is plentiful [164-176]. The most popular ones are the Akaike's Information Criterion (ACI) [164] and the Minimum Description Length (MDL) criterion [165]. Including a model order selection function into the expert system increases the autonomy of system identification.
- **Sampling interval selection** - Selection of the proper sampling interval, T_s, is crucial to the process of system identification because the quality of the estimated model depends upon it. The upper limit of the sampling interval is determined by the Nyquist interval, T_{Nq}. Oversampling must be avoided in order to maintain good accuracy in system identification. System identification using the ARMA model was found to be very sensitive to the selection of the sampling interval [170]. The expert system would be enhanced by a sampling interval selection functional.
- **Feedback loop delay calculation** - System response to control action can have a significant delay relative to the sampling interval. This delay originates from the system inertia in response to a control action. If this delay is significant it should be included in the system identification. During the experimental tests it was found that if an extra delay of one sample is included in the control signal for the system identification, the controller's performance improves. This delay estimation should be included as an expert system function in the future.

- **Robust system identification** - The concurrently occurring system identifications should consist of at least three different system identification techniques, where the techniques perform concurrently a few different versions of parametrization. For example, the RLS, FQR and IQR algorithms should be parametrized with λ_{MIN}, λ_{MED}, and λ_{MAX} parameters and similarly the LMS algorithms with μ_{MIN}, μ_{MED}, and μ_{MAX}. This way the number of concurrently performing algorithms would increase by three times, ensuring fast, medium and slow system identifications. Similarly, the parametrization of the KF with weighting matrices Q_n and R_n, would result in four to nine combinations for the KF implementation. Thereby the robustness of system identification would improve for all situations.
- **Weight matrix selection** - The H_∞ design is one of the most advanced techniques available today for a robust controller. However, one big drawback of this technique is the experience and skill needed to design the shape of the weighting functions on sensitivity and closed-loop transfer functions to achieve the desired goal [177]. The practical issue of the design of weighting matrices lacks a systematic approach. In this thesis each weighting matrix has been simplified by a coefficient that multiplies the unity matrix. During tests, it was found that the ratio between these coefficients determines the infimum for the H_∞ controller. Further research should be conducted to determine the possibility of a systematic algorithm for tuning these coefficients. If the weights selection were performed by an online expert system, the proposed H_∞ MPC could play an important role in industrial control.

In conclusion, a novel robust adaptive control design procedure has been introduced in this thesis, which integrates an H_∞ robust controller with a robust parameter estimation. This integration resulted in an H_∞ MPC algorithm, which has been formulated in a way to achieve a systematic, ready for application methodology for building an H_∞ MPC controller. The controller design has been simplified by translating the complex mathematical theory of control into a procedure that is easy to follow and implement. The developed theory contributes to a thorough understanding of the application of H_∞ MPC to industrial system control problems.

The effectiveness of the proposed procedure as a power system stabilizer has been demonstrated through simulation studies and by experimental tests. The successful implementation of the controller demonstrates how the H_∞ MPC can be effectively applied in an industrial design. The approach described is quite general and is, therefore, applicable to systems other than those studied in the context of this thesis.

13 References

[1] Kimura, H., "Chain-Scattering Approach to H_∞ Control", Birkhauser, Boston, 1997.

[2] Kalman, R. E., "Contribution to the theory of optimal control", Boletin de la Sociedad Matemetica Mexicana, Vol. 5, 1960, pp. 102-119.

[3] Kalman, R. E., "A new approach to linear filtering and prediction problems", Transactions of the ASME, Journal of Basic Engineering, Vol. 82D, March, 1960, pp. 35-45.

[4] Athans, M., and Falb, P. L., "Optimal Control - An Introduction to the Theory and its Applications", McGraw-Hill Book Company, New York, 1966.

[5] Athans, M., "The role and use of the stochastic linear-quadratic-gaussian problem in control system design", IEEE Transactions on Automatic Control, Vol. 16, No. 6, December, 1971, pp. 529-551.

[6] Chen, T., "A simple derivation of the H_2-optimal sampled-data controllers", North-Holland, System & Control Letters, Vol. 20, 1993, pp. 49-56.

[7] Green, M., and Limebeer, D. J. N., "Linear Robust Control", Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[8] Zames, G., "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses", IEEE Transactions on Automatic Control, Vol. 26, No. 2, April, 1981, pp. 301-320.

[9] Francis, B. A., and Zames, G., "On H_∞-optimal sensitivity theory for SISO feedback systems", IEEE Transactions on Automatic Control, Vol. 29, No. 1, January, 1984, pp. 9-16.

[10] Ball, J. A., and Cohen, N., "Sensitivity minimization in an H_∞ norm: Parametrization of all suboptimal solutions", International Journal of Control, Vol. 46, No. 3, September, 1987, pp. 785-816.

[11] Kimura, H., "Directional interpolation approach to H_∞- Optimization and robust stabilization", IEEE Transactions on Automatic Control, Vol. 32, No. 12, December, 1987, pp. 1085-1093.

[12] Doyle, J. C., Glover, K., Khargonekar, P. P., and Francis, B. A., "State-space solutions to standard H_∞ and H_∞ control problems", IEEE Transactions on Automatic Control, Vol. 34, No. 8, August, 1989, pp. 831-847.

[13] Green, M., Klover K., Limbeer, D. J. N., and Doyle, J. C., "A J-spectral factorization approach to H_∞ control", SIAM Journal on Control and Optimization, Vol. 28, 1990, pp. 1350-1371.

[14] Stoorvogel, A. A., and Trentelman, H. L., "Quadratic matrix inequality in singular H_∞ control with state feedback", SIAM Journal on Control and Optimization, Vol. 28, No. 5, September, 1990, pp. 1190-1208.

[15] Tadmor, G., "Worst case design in time domain: The maximum principle and the standard H_∞ problem", Mathematics of Control, Signals, and Systems, Vol. 3, No. 4, 1990, pp. 301-324.

[16] Gahinet, P., and Apkarian, P., "Linear matrix inequality approach to H_∞ control", International Journal of Robust and Nonlinear Control, Vol. 4, No. 4, July-August, 1994, pp. 421-448.

[17] Dragan, V., Halanay, A., and Ionescue, V., "Infinite horizon disturbance attenuation for discrete-time systems: A Popov-Yakubovich approach", Integral Equations and Operator Theory, Vol. 19, 1994, pp. 153-215.

[18] Feintuch, A., "Robust Control Theory in Hilbert Space", Springer-Verlag, New York, 1998.

[19] Basar, T., and Bernhard P., "H_∞ - Optimal Control and Related Minimax Design Problems - A Dynamic Game Approach", Second Edition, Birkhauser, Boston, 1995.

[20] Hassibi, B., Sayed, A. H., and Kailath, T., "Indefinite-Quadratic Estimation and Control", SIAM Applied and Numerical Mathematics, Philadelphia, 1999.

[21] de Callafon, R. A., “Feedback Oriented Identification for Enhanced and Robust Control”, PhD thesis, Delft University of Technology, Delft, The Netherlands, 1998.

[22] Camacho, E. F., and Bordons, C., “Model Predictive Control”, Second Edition, Springer-Verlag London Ltd., London, 2004.

[23] Rao, C. V., Wright, S. J., and Rawlings, J. B., “Application of interior-point methods to model predictive control”, Journal of Optimization Theory and Applications, Vol. 99, No. 3, December, 1998, pp. 723-757.

[24] Camponogara, E., Jia, D., Krogh, B. H., and Talukdar, S., “Distributed model predictive control”, IEEE Control Systems Magazine, Vol. 22, No. 1, February, 2002, pp. 44-52.

[25] Henson, M. A., and Seborg, D. E., “Nonlinear Process Control”, Prentice Hall, Upper Saddle River, New Jersey, 1997.

[26] Rácz, L. Z., and Bókay, B., “Power System Stability”, Akadémiai Kiadó, Budapest, 1988.

[27] Rogers, G., “Demystifying power system oscillations”, IEEE Computer Applications in Power, Vol. 9, No. 3, July, 1996, pp. 30-35.

[28] Pavella, M., and Murthy, P. G., “Transient Stability of Power Systems: Theory and Practice”, John Wiley & Sons, New York, 1994.

[29] Park, R. H., “Two-reaction theory of synchronous machines - Generalized methods of analysis: Part I ”, AIEE Transactions, Vol. 48, July, 1929, pp. 716–727.

[30] Ljung, L., "System Identification: Theory for the User", Prentice Hall - Information and System Science Series, Upper Saddle River, New Jersey, 1999.

[31] Forssell, U., "Closed-loop Identification – Methods, Theory, and Applications", PhD Thesis, Linkoping University, Linkoping, Sweden, 1999.

[32] Söderström, T., and Stoica, P., "System Identification", Prentice Hall, New York, 1989.

[33] Söderström, T., "Comments on 'Identification of closed-loop systems via least-squares method'", International Journal of Adaptive Control and Signal Processing, Vol. 13, No. 1, February, 1999, pp. 37-41.

[34] Forssell, U., and Ljung, L., "Closed-loop Identification Revisited", Automatica, Vol. 35, No. 7, July, 1999, pp. 1215-1241.

[35] Wang, L. Y., and Yin, G. G., "Persistent identification of systems with unmodeled dynamics and exogenous disturbances", IEEE Transactions on Automatic Control, Vol. 45, No. 7, July, 2000, pp. 1264-1256.

[36] Zhang, Y., and Feng, C., "Identification of closed-loop systems via least-squares method", International Journal of Adaptive Control and Signal Processing, Vol. 12, No.1, February, 1998, pp. 29-39.

[37] Johnson, C. R. Jr., "On the interaction of adaptive filtering, identification, and control", IEEE Signal Processing Magazine, Vol. 12, No. 2, March, 1995, pp. 22-37.

[38] Zhou, T., “Unfalsified plant model parametrization from closed-loop experimental data”, Automatica, Vol. 33, No. 5, May, 1997, pp. 805-820.

[39] Landau, I. D., “Identification in closed loop: a powerful design tool (better design models, simpler controllers)”, Control Engineering Practice, Vol. 9, No. 1, January, 2001, pp. 51-65.

[40] Hakvoort, R. G., “System Identification for Robust Process Control: Nominal Models and Error Bounds“, PhD Thesis, Delft University of Technology, Delft, The Netherlands, 1994.

[41] Landau, I. D., “From robust control to adaptive control”, Control Engineering Practice, Vol. 7, No. 9, September, 1999, pp. 1113-1124.

[42] Van den Hof, P., “Closed-loop issues in system identification”, Annual Reviews in Control, Vol. 22, 1998, pp. 173-186.

[43] Isermann, R., “Digital Control Systems: Stochastic Control, Multivariable Control, Adaptive Control, Applications”, Second Edition, Springer-Verlage, Berlin, Heidelberg, 1991.

[44] Narendra, K., and Cheng Xiang, “Adaptive Control of Discrete-Time Systems Using Multiple Models”, IEEE Transactions on Automatic Control, Vol. 45. No. 9, September, 2000, pp. 1669-1686.

[45] Autenrieth, T., and Rogers, E., “Performance enhancement for a class of multiple model adaptive control schemes”, International Journal of Adaptive Control and Signal Processing, Vol. 13, No. 2, March, 1999, pp. 105-127.

[46] Goodwin, G. C., Gevers, M., and Ninness, B., "Quantifying the error in estimated transfer functions with application to model order selection", IEEE Transactions on Automatic Control, Vol. 37, No. 7, July, 1992, pp. 913-928.

[47] Haykin, S., "Adaptive Filter Theory", Third Edition, Prentice Hall, Upper Saddle River, New Jersey, 1996.

[48] Van den Hof, P., "Closed-loop issues in system identification", Annual Reviews in Control, No. 22, 1998, pp. 173-186.

[49] Weiss, L., and Infante, E. F., "Finite time stability under perturbing forces and on product spaces", Differential Equations and Dynamical Systems, Proceedings of an International Symposium, Puerto Rico, Edited by Hale, J. K., and La Salle, J. P., Academic Press, New York, 1967, pp. 341-350.

[50] La Salle, J. P., and Lefschetz, S., "Stability by Liapunov's Direct Method - With Applications", Academic Press, New York, 1961.

[51] Safonov, M. G., "Focus on the knowable", in "Control Using Logic-Based Switching", Editor: Morse, A. S., Springer Verlag, Berlin, 1996, pp. 224-233.

[52] Kosut, R. L., "Uncertainty Model Unfalsification: A System Identification Paradigm Compatible with Robust Control Design", Proceedings of the 34th IEEE Conference on Decision and Control (Cat. No.95CH35803), Vol. 4, 1995, pp. 3492-3497.

[53] Wodoley, B. R., How, J. P., and Kosut, R. L., "Direct unfalsified controller design - solution via convex optimization", Proceedings of the 1999 American Control Conference (Cat. No.99CH36251), Vol. 5, 1999, pp. 3302-3306.

[54] Lewis, F. L., and Syrmos, V. L., “Optimal Control”, Second Edition, John Wiley & Sons, New York, 1995.

[55] Ninness, B., and Goodwin, G., “Estimation of Model Quality”, System Identification (SYSID 1994), A Postprint Volume from the IFAC Symposium, Vol. 1, 1995, pp. 25-44.

[56] Kay, S. M., “Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory”, Prentice Hall - Signal Processing Series, New Jersey, 1998.

[57] Golub, G. H., and Van Loan, C. F., “Matrix Computations”, Johns Hopkins University Press, Baltimore, 1996.

[58] Cadzow, J. A., “Minimum l_1, l_2 and l_∞ norm approximate solutions to an overdetermined system of linear equations”, Digital Signal Processing, Vol. 12, No. 4, October, 2002, pp. 524-560.

[59] Jacklin, S. A., “Comparison of five system identification algorithms for rotorcraft higher harmonic control”, NASA Center for Aerospace Information, NASA/TP-1998-207687, Springfiled, VA, May, 1998.

[60] Åström, K. J., and Wittenmark, B., “Adaptive Control”, Second Edition, Addison-Wesley Publishing Company, Massachusetts, 1995.

[61] Widrow, B., and Walach, E., “Adaptive Inverse Control”, Prentice Hall, Upper Saddle River, New Jersey, 1996.

[62] Fogel, E., and Huang, Y. F., “On the value of information in system identification-bounded noise case”, Automatica, Vol. 18, No. 2, March, 1982, pp. 229-238.

[63] Grewal, M. S., and Andrews, A. P., “Kalman Filtering: Theory and Practice”, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[64] Srinath, M. D., Rajasekaran, P. K., and Viswanathan, R., “Introduction to Statistical Signal Processing with Applications”, Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[65] Chen, G., “Approximate Kalman Filtering”, Word Scientific Publishing Co., River Edge, New Jersey, 1993.

[66] Sayed, A. H., and Kailath, T., “A state-space approach to adaptive RLS filtering”, IEEE Signal Processing Magazine, Vol. 11, No. 3, July, 1994, pp. 18-60.

[67] Zheng-She Liu, “QR methods of O(N) complexity in adaptive parameter estimation”, IEEE Transactions on Signal Processing, Vol. 43, No. 3, March, 1995, pp. 720-729.

[68] Alexander S. T., and Ghirnikar A. L., “A method for recursive least squares filtering based upon an inverse QR decomposition”, IEEE Transactions on Signal Processing, Vol. 41, No. 1, January, 1993, pp. 20-30.

[69] Zheng-She Liu, and Jian Li, “A QR-based least mean square algorithm for adaptive parameter estimation”, IEEE Transaction on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 45, No. 3, March, 1998, pp. 321-329.

[70] Aizerman, M., “On a problem concerning stability in the large dynamical systems”, Uspekhi Matematicheskiks Nauk, No. 4, 1949, pp. 187-188.

[71] Kalman, R. E., “Physical and mathematical mechanisms of instability in nonlinear automatic control systems”, Transactions of the ASME, Vol. 79, 1957, pp. 533-566.

[72] Vidyasagar, M., “Beyond H_∞-design: robustness, disturbance rejection and Aizerman-Kalman type conjectures in general signal spaces”, International Journal of Robust and Nonlinear Control, Vol. 10, No. 11-12, September-October, 2000, pp. 961-982.

[73] Doyle, J. C., Glover, K., Khargonekar, P. P., and Francis, B. A., “State-space solutions to standard H_2 and H_∞ control problems”, IEEE Transactions on Automatic Control, Vol. 34, No. 8, August, 1989, pp. 831-847.

[74] Iglesias, P. A., and Glover, K., “State space approach to discrete-time H_∞ control”, International Journal of Control, Vol. 54, No. 5, November, 1991, pp. 1031-1073.

[75] Stoorvogel, A. A., “The discrete time H_∞ control problem with measurement feedback”, SIAM Journal on Control and Optimization, Vol. 30, No. 1, January, 1992, pp. 182-202.

[76] Stoorvogel, A. A., Saberi, A., and Chen, B. M., “The discrete-time H_∞ control problem with measurement feedback”, International Journal of Robust and Nonlinear Control, Vol. 4, No. 4, July-August, 1994, pp. 457-479.

[77] Ionescu, V., and Weiss, M., "Two-Riccati formulae for the discrete-time H_∞ -control problem", International Journal of Control, Vol. 57, No. 1, January, 1993, pp. 141-195.

[78] Stoorvogel, A. A., "The H_∞ Control Problem: A State Space Approach", Prentice Hall, New York, 1992.

[79] Chen, T., and Francis, B., "Optimal Sampled-Data Control Systems", Springer-Verlag London Ltd., London, 1995.

[80] Zhou, K., Doyle, J. C., and Glover, K., "Robust and Optimal Control", Prentice Hall, Upper Saddle River, New Jersey, 1996.

[81] Chen, B. M., "H_∞ Control and Its Applications", Springer-Verlag London Ltd., London, 1998.

[82] Tay, T., Mareels, I., and Moor, J. B., "High Performance Control", Birkhäuser, Boston, 1998.

[83] Zhou, K., and Doyle, J. C., "Essentials of Robust Control", Prentice Hall, Upper Saddle River, New Jersey, 1998.

[84] Helton, J. W., and James, M. R., "Extending H_∞ Control to Nonlinear Systems - Control of Nonlinear Systems to Achieve Performance Objective", SIAM, Philadelphia, 1999.

[85] Basar, T., and Olsder, G. J., “Dynamic Noncooperative Game Theory”, Second Edition, SIAM, Philadelphia, 1999.

[86] Rangan, S., “Validation, Identification and Control of Robust Control Uncertainty Models”, PhD Thesis, Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, 1997.

[87] van der Schaft, A. J., “L_2-gain analysis of nonlinear systems and nonlinear state-feedback H_inf control”, IEEE Transactions on Automatic Control, Vol. 37, No. 6, June, 1992, pp. 770-784.

[88] Iglesias, P. A., “An entropy formula for time-varying discrete-time control system”, SIAM Journal of Control and Optimization, Vol. 34, No. 5, September, 1996, pp. 1691-1706.

[89] Trentelman, H. L., Stoorvogel, A. A., and Hautus, M., “Control Theory for Linear Systems”, Springer-Verlag London Ltd., London, 2001.

[90] Amato, F., Mattei, M., and Pironti, A., “Guaranteeing cost strategies for linear quadratic differential games under uncertain dynamics”, Automatica, Vol. 38, No. 3, March, 2002, pp. 507-515.

[91] McEneaney, M. W., and Fitzpatrick, B., “Control for UAV operations under imperfect information”, The American Institute of Aeronautics and Astronautics Inc., AIAA-2002-3418, 2002.

[92] James, M. R., “On the certainty equivalence principle and the optimal control of partially observed dynamic games”, IEEE Transactions on Automatic Control, Vol. 39, No. 11, November, 1994, pp. 2321-2324.

[93] Basar, T., “A dynamic games approach to controller design: disturbance rejection in discrete-time”, IEEE Transactions on Automatic Control, Vol. 36, No. 8, August, 1991, pp. 936-952.

[94] Juniper, J., “Macroeconomic stability and control in an uncertain word”, University of South Australia, Web: business2.unisa.edu.au, CoBAR Working Paper Series, 2000-06, Australia, Adelaide, 2000.

[95] Magni, L., Nijmeijer, H., van der Schaft, A. J., “A receding-horizon approach to the nonlinear H_inf control problem”, Automatica, Vol. 37, No. 3, March, 2001, pp. 429-435.

[96] Lee, L-H., Goodwin, G., and Kolodziej, W., “Interconnections between continuous and discrete games with applications to H_∞ ”, Proceedings of the 29th IEEE Conference on Decision and Control (Cat. No.90CH2917-3), Vol. 4, 1990, pp. 2425-2430.

[97] James, M. R., Baras, J. S., and Elliott, R. J., “Risk-sensitive control and dynamic games for partially observed discrete-time nonlinear systems”, IEEE Transactions on Automatic Control, Vol. 39, No. 4, April, 1994, pp. 780-792.

[98] Van De Water, H., and Willems, J. C., “The certainty equivalence property in stochastic control theory”, IEEE Transactions on Automatic Control, Vol. 26, No. 5, October, 1981, pp. 1080-1087.

[99] Gonzalez-Trejo, J. I., Hernandez-Lerma, O., and Hoyos-Reyes, L. F., "Minimax control of discrete-time stochastic systems", SIAM Journal on Control and Optimization, Vol. 41, No. 5, 2002, pp. 1626-1659.

[100] Brozenec, T. F., Tsao, T. C., and Safonov, M. G., "Controller validation", International Journal of Adaptive Control and Signal Processing, Vol. 15, No. 5, August, 2001, pp. 431-444.

[101] Stoorvogel, A. A., "Numerical problems in robust and H_∞ optimal control", NICONET report, 1999-13, Belgium, 1999.

[102] Balas, J. G., Doyle, C. J., Glover, K., Packard, A., and Smith, R., "μ-Analysis and Synthesis Toolbox for Use with MATLAB", User's Guide Version 3, The MatWorks Inc., 1998.

[103] Tchernychev, A., and Sideris, A., "Robust discrete-time control design with time-varying parametric uncertainties", International Journal of Robust and Nonlinear Control, No. 9, No. 4, April, 1999, pp. 199-213.

[104] Wie, B., and Bernstein, D. S., "A benchmark problem for robust control design", Proceedings of the 1990 American Control Conference (IEEE Cat. No.90CH2896-9), Vol. 1, 1990, pp. 961-962.

[105] Matlab, "Robust Control Toolbox User's Guide", The MatWorks Inc., 2001.

[106] Fontes, F. A. C. C., “A general framework to design stabilizing nonlinear model predictive controllers”, Systems & Control Letters, Vol. 42, No. 2, February, 2001, pp. 127-143.

[107] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., “Constrained model predictive control: stability and optimality”, Automatica, Vol. 36, No. 6, June, 2000, pp. 789-814.

[108] Blanchini, F., “Set invariance in control”, Automatica, Vol. 35, No. 11, November, 1999, pp. 1747-1767.

[109] De Schutter, B., and van den Boom, T., “Model predictive control for max-plus-linear discrete event systems”, Automatica, Vol. 37, No. 7, July, 2001, pp. 1049-1056.

[110] Camacho, E. F., and Bordons, C., “Model Predictive Control in the Process Industry”, Springer-Verlag, London, 1995.

[111] Ki Baek Kim, Jae-Won Lee, and Wook Hyun Kwon, “Intervalwise receding horizon H_inf-tracking control for discrete linear periodic systems”, IEEE Transactions on Automatic Control, Vol. 45, No. 4, April, 2000, pp. 747-752.

[112] Lall, S., and Glover, K., “Robust performance and adaptation using receding horizon H_inf control of time varying systems”, Proceeding of the 1995 American Control Conference (IEEE Cat. No.95CH35736), Vol. 3, 1995, pp. 2384-2388.

[113] Beard, R. W., and McLain, T. W., “Successive Galerkin approximation algorithms for nonlinear optimal and robust control”, International Journal of Control, Vol. 71, No. 5, November, 1998, pp. 717-743.

[114] Tsiotras, P., Corless, M., and Rotea, M. A., "An L_2 disturbance attenuation solution to the nonlinear benchmark problem", International Journal of Robust and Nonlinear Control, Vol. 8, No. 4-5, April, 1998, pp. 311-330.

[115] Lin, W., and Xie, L. "A link between H_inf control of a discrete-time nonlinear system and its linearization", International Journal of Control, Vol. 69, No. 2, January, 1998, pp. 301-314.

[116] Magni, L., De Nicolao, G., Scattolini, R., and Allgower, F., "Robust model predictive control for nonlinear discrete-time systems", International Journal of Robust and Nonlinear Control, Vol. 13, No. 3-4, March-April, 2003, pp. 229-246.

[117] Rao, C. V., Rawlings, J. B., and Lee, J. H., "Constrained linear state estimation - a moving horizon approach", Automatica, Vol. 37, No. 10, October, 2001, pp. 1619-1628.

[118] Chen, G. P., Malik, O. P., and Hope, G. S., "Generalized discrete control system design method with control limit consideration", IEE Proceedings - Control Theory Applications, Vol. 141, No.1, January, 1994, pp. 39-47.

[119] Chen, G. P., and Malik, O. P., "Tracking constrained adaptive power system stabilizer", IEE Proceedings - Generation, Transmission and Distribution, Vol. 142, No. 2, March, 1995, pp. 149-156.

[120] Gokaraju, R., "Beyond Gain-Type Scheduling Controllers: New Tools of Identification and Control for Adaptive PSS", PhD Thesis, University of Calgary, Calgary, 2000.

[121] Mortlock, J. R., and Davies, H. M. W., “Power System Analysis”, Chapman and Hall, London, England, 1952.

[122] deMello, F. P., and Laskowski, T. F., “Concept of power system dynamic stability”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-94, No. 3, May-June, 1975, pp. 827-883.

[123] Larsen, E. V. and Swann, D. A., “Applying power system stabilizer: Parts 1-3”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 6, June, 1981, pp. 3017-3046.

[124] Ghosh, A., Ledwich, G., Malik, O. P., and Hope, G. S., “Power system stabilizer based on adaptive control techniques”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No. 8, August, 1984, pp. 1983-1989.

[125] Cheng, S. J., Malik, O. P., and Hope, G. S., “Damping of multi-modal oscillations in power systems using a dual-rate adaptive stabilizer”, IEEE Transactions on Power Systems, Vol. 3, No. 1, February, 1988, pp. 101-108.

[126] Stevenson, W. D. Jr., “Elements of Power System Analysis”, McGraw-Hill, New York, 1982.

[127] Yu, Y. N., "Electric Power System Dynamics", Academic Press, New York, 1983.

[128] Girgis, G., “WECC Power System Stabilizer Tuning Guidelines”, Western Electricity Coordination Council, Salt Lake City, UT, Web www.wecc.biz, May, 2002.

[129] Hariri, A., “An Adaptive Fuzzy Logic Power System Stabilizer”, PhD Thesis, University of Calgary, Calgary, 1996.

[130] Chen, S., “Power System Uncertainty Analysis And Robust Stabilizer Design”, PhD Dissertation, University of Calgary, Calgary, 1995.

[131] Summers, C. M., “The Faraday electrical machines laboratory”, Proceedings of the IEEE, Vol. 64, No. 11, November, 1976, pp. 1556-1582.

[132] Huber, D. W., Hope, G. S., and Malik, O.P., “Synchronous machine field time constant regulator”, Manual in the Department of Electrical Engineering, The University of Calgary, Calgary,

[133] Hingorani, N. G., “Power electronics in electric utilities: role of power electronics in future power systems”, Proceedings of the IEEE, Vol. 76, No. 4, April, 1988, pp. 481-482.

[134] Gilany, M., “A Microprocessor Based Relay for Parallel Transmission Line”, A Thesis, University of Calgary, Calgary, 1992.

[135] Spectrum Digital Incorporated, “TMS320C6713 DSK Technical Reference”, Literature Number: 506735-0001, Stafford, Texas, 2004.

[136] Texas Instruments Incorporated, “TMS320C6711 DSK Help”, Provided in soft form with the TMDS320006711 kit on CD, Dallas, Texas, 2000.

[137] Texas Instruments Incorporated, “Code Composer Studio Getting Started Guide”, Literature Number: SPRU509, Dallas, Texas, 2001.

[138] Texas Instruments Incorporated, “TMS320 DSP/BIOS User's Guide, SPRU423, Dallas, Texas, 2002.

[139] Texas Instruments Incorporated, “TMS320C6000 Assembly Language Tools User's Guide, SPRU186, Dallas, Texas, 2002.

[140] Texas Instruments Incorporated, “TMS320C6000 DSP/BIOS Application Programming Interface (API) Reference Guide”, Literature Number: SPRU403, Dallas, Texas, 2002.

[141] Texas Instruments Incorporated, “TMS320C6000 Optimizing Compiler User's Guide”, Literature Number: SPRU187, Dallas, Texas, 2002.

[142] Texas Instruments Incorporated, “TMS320C6000 Programmer's Guide”, Literature Number: SPRU198, Dallas, Texas, 2002.

[143] Texas Instruments Incorporated, “TMS320C6000 Chip Support Library API Reference Guide”, Literature Number: SPRU401, Dallas, Texas, 2003.

[144] Texas Instruments Incorporated, “TMS320C6000 DSK Board Support Library API User's Guide”, Literature Number: SPRU432, Dallas, Texas, 2001.

[145] Texas Instruments Incorporated, “TMS320C6000 CPU and Instruction Set Reference Guide”, Literature Number: SPRU189, Dallas, Texas, 2000.

[146] Texas Instruments Incorporated, “TMS320C6000 Peripherals Reference Guide”, Literature Number: SPRU190, Dallas, Texas, 2001.

[147] Kay, S. M., “Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory”, Prentice Hall - Signal Processing Series, New Jersey, 1993.

[148] Papoulis, A., and Pillai, S. U., “Probability, Random Variables, and Stochastic Processes”, Fourth Edition, McGraw Hill, New York, 2002.

[149] Cramér, H., “Mathematical Methods of Statistics”, Princeton University Press, Princeton, New Jersey, 1958.

[150] Mitra, S. K., “On the probability distribution of the sum of uniformly distributed random variables”, SIAM Journal on Applied Mathematics, Vol. 20, No. 2, March, 1971, pp. 195-198.

[151] Andersson, G., “Dynamics and Control of Electric Power Systems”, Lectures - Swiss Federal Institute of Technology Zurich - Power Systems Laboratory, Zurich March, 2003.

[152] The Generation Interconnection Development Team, “Technical Requirements for the Interconnection of Generation Resources”, U.S. Department of Energy Bonneville Power Administration Transmission Business Line, Vancouver, Washington, April, 1999.

[153] Adkins, B., and Harley, R. G., “The General Theory of Alternating Current Machines: Application to Practical Problems”, Chapman and Hall, London, England, 1975.

[154] Phillips, C. L., and Troy Nagle H. Jr., “Digital Control System Analysis and Design”, Third Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[155] Andersson, G., “Dynamics and Control of Electric Power Systems”, EEH - Power Systems Laboratory, Swiss Federal Institute of Technology, Zurich, March, 2003.

[156] Larsen, E. V., and Swann, D. A., “Applying power system stabilizers: Parts 1-3”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 6, June, 1981, pp. 3017-3046.

[157] Cai, L.J. and Erlich, I., “Simultaneous coordinated tuning of PSS and FACTS controller for damping power system oscillations in multimachine systems”, 2003 IEEE Bologna Power Tech (IEEE Cat. No.03EX719), Vol. 2, 2003.

[158] Hiskens, I. A., “Systematic tuning of nonlinear power system controllers”, Proceedings of the 2002 IEEE International Conference on Control Applications (Cat. No.02CH37330), Vol. 1, No. 1, 2002, pp. 19-24.

[159] Long Term Forecasts & Assessments Department, “Connection Assessment & Approval Process”, Independent Electricity Market Operator,
Web: www.theIMO.com, CAA ID 2002-069, Toronto, February 14, 2003.

[160] Mithulananthan, N., Cañizares, C. A., and Reeve, J., “Tuning, performance and interactions of PSS and FACTS Controllers”, 2002 IEEE Power Engineering Society Summer Meeting (Cat. No.02CH37376), Vol. 2, No. 2, 2002, pp. 981-987.

[161] Rafee, N., “Optimal Multirate Discretization of Analog Controllers: Theory and Application”, PhD Thesis, University of Calgary, Calgary, 1997.

[162] IEEE Excitation System Model Working Group, “Excitation System Models for Power System Stability Studies”, Draft 15 for ANSI/IEEE Standard P421.5/D15, 1990.

[163] Eitzmann, M., “Excitation System Tuning and Testing for Increased Power System Stability”, GE Energy BROC10605, 2004.

[164] Akaike, H., “A new look at the statistical model identification”, IEEE Transactions on Automatic Control, Vol. 19, No. 6, December, 1974, pp. 716-723.

[165] Rissanen, J., “Modelling by shortest data description”, Automatica, Vol. 14, No. 5, September, 1978, pp. 465-471.

[166] Rohrs, C. E., Valavani, L., Athans, M., and Stein, G., “Robustness of continuos-time adaptive control algorithms in the presence of unmodeled dynamics”, IEEE Transactions on Automatic Control, Vol. 30, No. 9, September, 1985, pp. 881-889.

[167] Liang, G., Wilkes, D. M., and Cadzow, J. A., “ARMA model order estimation based on the eigenvalues of the covariance matrix”, IEEE Transactions on Signal Processing, Vol. 41, No. 10, October, 1993, pp. 3003-3009.

[168] Kay, S., “Conditional model order estimation”, IEEE Transactions on Signal Processing, Vol. 49, No. 9, September, 2001, pp. 1910-1917.

[169] Djuric, P. M., “Asymptotic MAP criteria for model selection”, IEEE Transactions on Signal Processing, Vol. 46, No. 10, October, 1998, pp. 2726-2735.

[170] Smail, M., Thomas, M., and Lakis, A., "ARMA models for modal analysis: effect of model orders and sampling frequency", Mechanical Systems and Signal Processing, Vol. 13, No. 6, November, 1999, pp. 925-941.

[171] Smail, M., Thomas, M., and Lakis, A., "Assessment of optimal ARMA model orders for modal analysis", Mechanical Systems and Signal Processing, Vol. 13, No. 5, September, 1999, pp. 803-819.

[172] Guyon, X., and Yao, J., "On the underfitting and overfitting sets of models chosen by order selection criteria", Journal of Multivariante Analysis, Vol. 70, 1999, pp. 221-249.

[173] Goodwin, G. C., Gevers, M., and Ninness, B., "Quantifying the error in estimated transfer functions with application to model order selection", IEEE Transactions on Automatic Control, Vol. 37, No. 7, July, 1992, pp. 913-928.

[174] Moddemeijer, R., and Spaanenberg, B., "Selection of sufficient stochastic data-models applied to modelling", IEEE ProRISC99, 1999.

[175] Pan, J., and Levine, W. S., "Theoretical and experimental study of order estimation", University of Maryland - System Research Centre, Technical Research Report, Maryland, 1992.

[176] Stoica, P., and Jansson, M., "Transfer function approach to MIMO system identification", Proceedings of the IEEE Conference on Decision and Control, Vol. 3,1999, pp. 2400-2405.

[177] Kruck, K., and Kelkar, A. G., “Development of robust control strategies for aero-space systems - robust control of elastic systems with applications to spacecraft and terrestrial structures”, Iowa State University, Final Report for Iowa Space Grant Consortium, “Seed” Grant - PY13, December, 2002.

Appendix A - Single-Machine Power System

The structural diagram of the single-machine infinite-bus power system model is shown in Figure 8-1. Each interconnected block is described by linear/nonlinear differential equations. The signal to be controlled is the power angle, denoted by δ. The PSS has access to the rotor speed, $\omega(t)$, or to the generator electric power, $P_e(t)$, and produces an auxiliary control signal, $U_{pss}(t)$, applied to the automatic voltage regulator (AVR).

A.1 Generator Model

The block diagram of a generator's mathematical model is shown in Figure A-1, where the external signals T_m and E_0 are the driving mechanical torque and infinite-bus voltage respectively.

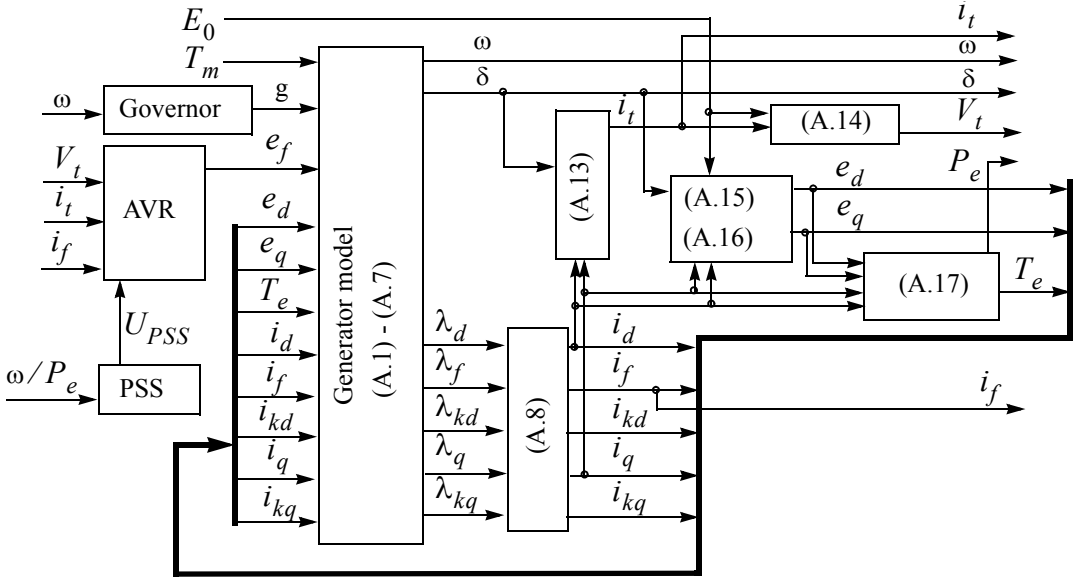


Figure A-1 Mathematical model for the power system

The generating unit is modelled by seven first order differential equations [153] [120][161]

$$\dot{\delta} = \omega_0 \omega \tag{A.1}$$

$$\dot{\omega} = \frac{1}{2H}[T_m + g + K_d\omega - T_e] \tag{A.2}$$

$$\dot{\lambda}_d = e_d + r_a i_d + \omega_0(1+\omega)\lambda_q \tag{A.3}$$

$$\dot{\lambda}_q = e_q + r_a i_q - \omega_0(1+\omega)\lambda_d \tag{A.4}$$

$$\dot{\lambda}_f = e_f - r_f i_f \tag{A.5}$$

$$\dot{\lambda}_{kd} = -r_{kd} i_{kd} \tag{A.6}$$

$$\dot{\lambda}_{kq} = -r_{kq} i_{kq}\,. \tag{A.7}$$

Where: ω_0 - nominal rotational speed, H inertia constant, K_d is the damping coefficient (representing the mechanical as well as the electrical damping effect), r_a - armature resistance, r_f - field resistance, r_{kd} - direct-axis damper resistance and r_{kq} - quadrature-axis damper resistance.

The relationship between flux linkages, (λ), and the corresponding currents, (i), is resolved in $d-q$ components

$$\begin{bmatrix} i_d \\ i_f \\ i_{kd} \end{bmatrix} = Y_d \begin{bmatrix} \lambda_d \\ \lambda_f \\ \lambda_{kd} \end{bmatrix} \qquad \begin{bmatrix} i_q \\ i_{kq} \end{bmatrix} = Y_q \begin{bmatrix} \lambda_q \\ \lambda_{kq} \end{bmatrix}. \tag{A.8}$$

The system admittances, Y_d and Y_q, are defined as follows

$$Z_d = x_{md}^2(x_d + x_f + x_{kd} - 2x_{md}) - x_d x_f x_{kd} \tag{A.9}$$

$$Z_q = x_{md}^2 - x_q x_{kq} \tag{A.10}$$

$$Y_d = \frac{1}{Z_d}\begin{bmatrix} x_f x_{kd} - x_{md}^2 & -x_{md}(x_{kd} - x_{md}) & -x_{md}(x_f - x_{md}) \\ x_{md}(x_{kd} - x_{md}) & -x_d x_{kd} + x_{md}^2 & x_{md}(x_d - x_{md}) \\ x_{md}(x_f - x_{md}) & x_{md}(x_d - x_{md}) & -x_f x_d + x_{md}^2 \end{bmatrix} \tag{A.11}$$

$$Y_q = \frac{1}{Z_q}\begin{bmatrix} x_{kq} & -x_{mq} \\ x_{mq} & -x_q \end{bmatrix}. \tag{A.12}$$

Where: x_f - field leakage reactance, x_{md} - direct-axis magnetizing reactance, x_{mq} - quadrature-axis magnetizing reactance, x_a - leakage reactance, $x_d = x_{md} + x_a$ - direct-axis synchronous reactance, $x_q = x_{mq} + x_a$ - quadrature-axis synchronous reactance, x_{kd} direct-axis damper leakage reactance and x_{kq} quadrature-axis damper leakage reactance.

The terminal current, i_t, is calculated from its direct and quadrature components, i_d and i_q, and from the power angle, δ,

$$i_t = (i_d \sin\delta + i_q \cos\delta) + j(i_q \sin\delta + i_d \cos\delta). \tag{A.13}$$

For the single-machine infinite-bus system the terminal voltage, V_t, is computed as

$$V_t = E_0 + (r_t + jx_t) i_t \tag{A.14}$$

where r_t and x_t are the transmission-line resistance and reactance, and E_0 is the infinite-bus voltage.

The transmission network with an equivalent impedance of $r_e - jx_e$, which connects the generator to an infinite-bus of voltage E_0, is given by the equations

$$e_d = E_0 \sin(\delta) + r_e i_d - x_e i_q \tag{A.15}$$

$$e_q = E_0 \cos(\delta) + r_e i_q + x_e i_d . \tag{A.16}$$

The electrical components of the machine torque, T_e, are expressed as

$$P_e = e_d i_d + e_q i_q \qquad T_e = P_e + r_a i_d^2 + r_a i_q^2 . \tag{A.17}$$

Parameter values for these equations are listed in Table A-1.

Table A-1 Generator parameters used in simulation

$H = 4.0$	$K_d = -0.0027$	$r_a = 0.007$	$r_f = 0.00089$
$r_{kd} = 0.023$	$r_{kq} = 0.023$	$r_e = 0.05$	$x_e = 0.6$
$x_d = 1.24$	$x_f = 1.33$	$x_{kd} = 1.15$	$x_{md} = 1.126$
$x_q = 0.743$	$x_{kq} = 0.652$	$x_{mq} = 0.626$	

All resistances/reactances are expressed in PER UNIT.

A.2 IEEE ST1A AVR and Exciter

The AVR and exciter models used in the system are from the IEEE Standard P421.5/ D15, Type ST1A, shown in Figure A-2 [162].

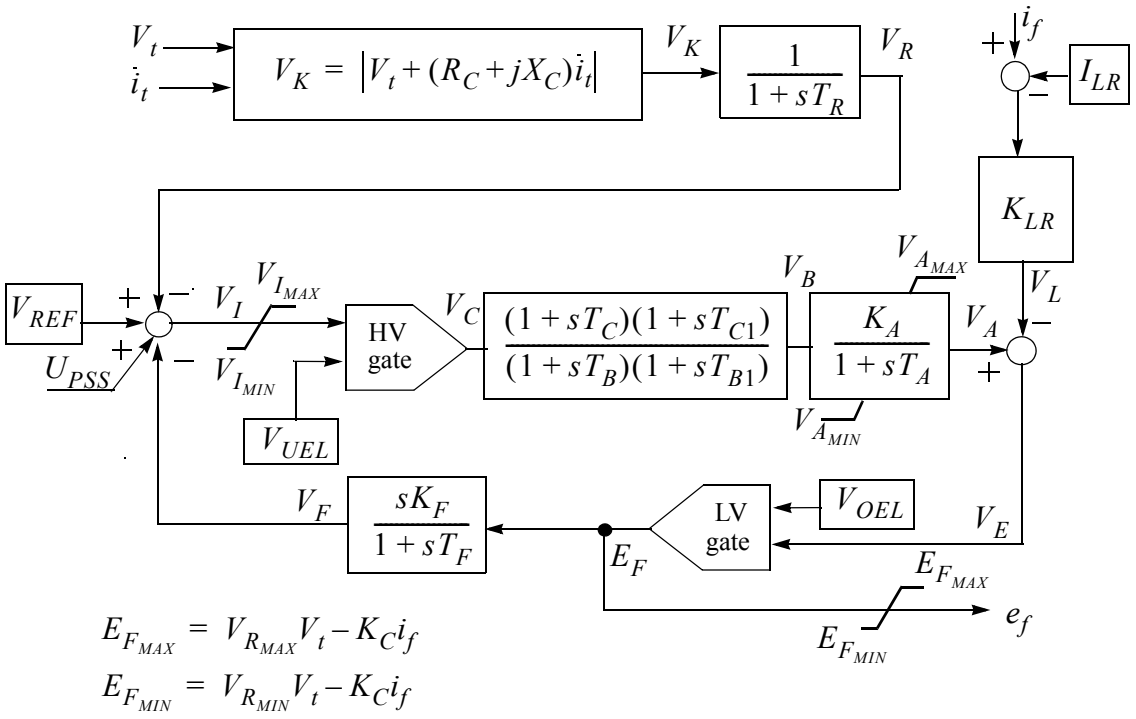


Figure A-2 AVR and exciter model

The symbols in Figure A-2 are defined as follows: V_t is the terminal voltage, i_t is the terminal current, i_f is the generator field winding current, I_{LR} is the exciter output current limit reference, V_{REF} is the AVR reference voltage, U_{PSS} is the power system stabilizer signal, V_{OEL} is the over-excitation limiter output, and V_{UEL} is the under-excitation limiter output. The HV gate and the LV gate select the larger and the smaller values from the two inputs respectively.

The AVR's control action is determined by the lag-lead compensator with time constants T_B, T_C, T_{B1} and T_{C1} and by the voltage regulator of proportional and integral action of time constant T_A and gain K_A. The time constants for the lag-lead compensator

should be selected such that $T_C > T_B$ for lead, and $T_{C1} < T_{B1}$ for lag type compensations. The local control loop is closed by the proportional-derivative action block with K_F and T_F of gain and time constant respectively.

Excitation systems with high gain and fast response times greatly aid transient stability (synchronizing torque) but at the same time tend to reduce small signal stability (damping torque) [163]. Consequently, for increased system stability an additional control loop is required, which is the so called power system stabilizer.

Table A-2 AVR and exciter parameters used in simulation

$R_C = 0.0$	$X_C = 0.0$	$T_R = 0.040$	$K_A = 190.0$
$K_C = 0.080$	$K_F = 0.05$	$T_A = 0.01$	$T_B = 10.0$
$T_C = 1.0$	$T_{B1} = 0.0$	$T_{C1} = 0.0$	$T_F = 1.0$
$V_{I_{MIN}} = -999$	$V_{I_{MAX}} = 999$	$V_{A_{MIN}} = -999$	$V_{A_{MAX}} = 999$
$V_{R_{MIN}} = -6.7$	$V_{R_{MAX}} = 7.8$	$K_{LR} = 0.0$	$I_{LR} = 0.0$

All resistances/reactances/voltages are expressed in PER UNIT. The time constants are shown in seconds.

A.3 IEEE PSS1A Conventional PSS

The objective of the PSS is to increase the damping of generator's rotor angle swings, which can occur in a broad range of frequencies in the power system [163]. The PSS provides a supplemental control signal that improves dynamic stability by increasing damp-

ing of power swing oscillations. The conventional power system stabilizer is a PSS1A type controller, from IEEE Standard P421.5/D15, shown in Figure A-3.

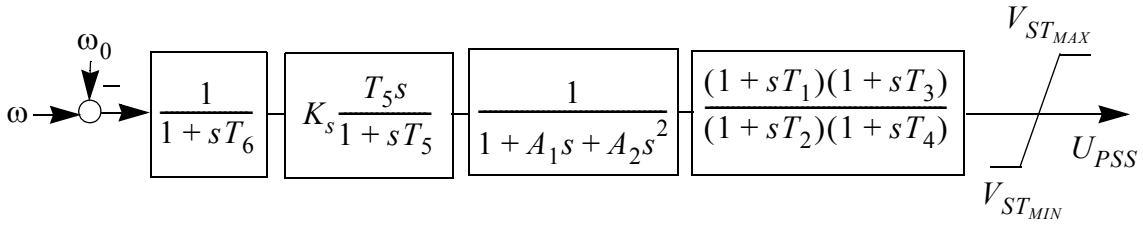


Figure A-3 IEEE standard power system stabilizer

PSS acts through the AVR excitation system, where it introduces a component of additional damping torque proportional to speed change.

The transfer function of the PSS1A consists of a low-pass input filter with T_6 time constant, presented in the first block. The second block is a high-pass filter, a derivative type regulator block, with T_5 time constant and K_s stabilizer gain. These two blocks can be considered as a proportional-integral-differential controller. In the third block, a second order torsional filter with A_1 and A_2 parameters is presented. Lastly, a lead-lag compensator with time constants of T_1, T_2, T_3 and T_4 is shown in the fourth block. The lead-lag blocks provide the appropriate phase-lead characteristics to compensate the phase lag between the exciter input and the generator's electrical torque. The time constants for the lead-lag compensator should be selected such that $T_1 > T_2$ for lead and $T_3 < T_4$ for lag type compensations.

Table A-3 CPSS parameters used in simulation

T_1 = 0.10	T_2 = 0.01	T_3 = 0.12	T_4 = 0.02

Table A-3 CPSS parameters used in simulation

$T_5 = 1.65$	$T_6 = 0.005$	$A_1 = 0.0$	$A_2 = 0.0$
$V_{ST_{MIN}} = -0.1$	$V_{ST_{MAX}} = 0.1$	$K_S = 0.11$	

All voltages are expressed in PER UNIT and the time constants are shown in seconds.

A.4 Governor Transfer Function

If a generator is embedded in a large interconnected system, it can with a very good approximation be modelled as connected to an infnite-bus, as presented in Figure A-4. In steady state the frequency is given by the one of the infnite-bus, ω_0. The governor controls only the power of the generator, not the frequency.

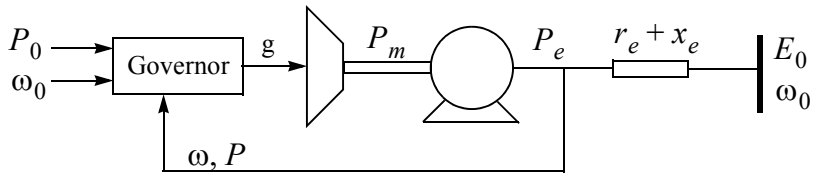


Figure A-4 Governor in a power system

The governor used in the system for simulation has a transfer function

$$g(s) = \left[a_g + \frac{b_g}{1 + sT_g}\right]\omega(s) \quad . \tag{A.18}$$

Parameter values for the equations are listed in Table A-4.

Table A-4 Governor parameters used in simulation

$a_g = -0.00133$	$b_g = -0.17$	$T_g = 0.25$

Appendix B - Multi-Machine Power System Model

The generating unit is modeled by five first order differential equations [130], [120]. These equations are presented below. The different transfer functions, which are also elements of the power system, are described with the single-machine infinite-bus model in Appendix A.

B.1 Generator Model

$$\dot{\delta} = \omega_0 \omega \tag{B.1}$$

$$\dot{\omega} = \frac{1}{2H}(T_m + g + K_d\omega - T_e) \tag{B.2}$$

$$T_{d0}'\dot{e}_q' = e_f - (x_d - x_d')i_d - e_q' \tag{B.3}$$

$$T_{d0}''\dot{e}_q'' = [e_q' - (x_d' - x_d'')i_d - e_q''] + T_{d0}''\dot{e}_q' \tag{B.4}$$

$$T_{q0}''\dot{e}_d'' = (x_q - x_q'')i_q - e_d''. \tag{B.5}$$

The structural diagram of the five machine power system is given in Figure 9-1, for which the parameters are defined below [120][129].

Table B-1 Generator parameters

	G_1	G_2	G_3	G_4	G_5
x_d	0.1026	0.1026	1.0260	0.1026	1.0260
x_q	0.0658	0.06580	0.658	0.0658	0.6580
x_d'	0.0339	0.0339	0.3390	0.0339	0.3390

Table B-1 Generator parameters

	G_1	G_2	G_3	G_4	G_5
x_d"	0.0269	0.0269	0.2690	0.0269	0.2690
x_q"	0.0335	0.0335	0.3350	0.0335	0.3350
T_{d0}'	5.6700	5.6700	5.6700	5.6700	5.6700
T_{d0}"	0.6140	0.6140	0.6140	0.6140	0.6140
T_{q0}"	0.7230	0.7230	0.7230	0.7230	0.7230
H	80.000	80.000	10.000	80.000	10.000

All resistances and reactances are expressed in PER UNIT and the time constants are shown in seconds.

Table B-2 AVR and simplified ST1A exciter parameters

	G_1	G_2	G_3	G_4	G_5
T_r	0.0400	0.0400	0.0400	0.0400	0.0400
K_a	190.00	190.00	190.00	190.00	190.00
K_c	0.0800	0.0800	0.0800	0.0800	0.0800
T_b	10.000	10.000	10.000	10.000	10.000
T_c	1.0000	1.0000	1.0000	1.0000	1.0000

All resistances and reactances are expressed in PER UNIT and the time constants are shown in seconds.

Table B-3 Governor parameters

	G_1	G_2	G_3	G_4	G_5
T_g	0.2500	0.2500	0.2500	0.2500	0.2500
a_g	-0.00015	-0.00015	-0.00133	-0.00015	-0.00133
b_g	-0.01500	-0.01500	-0.17000	-0.01500	-0.17000

All resistances and reactances are expressed in PER UNIT and the time constants are shown in seconds.

Table B-4 Transmission line parameters

Bus #	R_t	X_t	$B_t/2$
1 -- 7	0.00435	0.01067	0.01536
2 -- 6	0.00213	0.00468	0.00404
3 -- 6	0.01002	0.03122	0.03204
3 -- 6	0.01002	0.03122	0.03204
4 -- 8	0.00524	0.01184	0.01765
5 -- 6	0.00711	0.02331	0.02732
6 -- 7	0.04032	0.12785	0.15858
7 -- 8	0.01724	0.04153	0.06014

All resistances and reactances are shown in PER UNIT.

B.2 Operating Conditions and Loads for Operating Point #1

Table B-5 Power flow parameters

	G_1	G_2	G_3	G_4	G_5
P	5.1076	8.5835	0.8055	8.5670	0.8501
Q	6.8019	4.3836	0.4353	4.6686	0.2264
V	1.0750	1.0500	1.0250	1.0750	1.0250
δ	0.0000	0.3167	0.2975	0.1174	0.3051

All electrical power and voltages are expressed in PER UNIT and the power angles are in rad.

Table B-6 Load parameters

L_1	L_2	L_3
7.5 - j5.0	8.5 - j5.0	7.0 - j4.5

All resistances and reactances are shown in PER UNIT.

B.3 Operating Conditions and Loads for Operating Point #2

Table B-7 Power flow parameters for 2nd operating point

	G_1	G_2	G_3	G_4	G_5
P	3.1558	3.8835	0.4055	4.0670	0.4501
Q	2.9260	1.4638	0.4331	2.1905	0.2574
V	1.0500	1.0300	1.0250	1.0500	1.0250
δ	0.0000	0.1051	0.0943	0.0361	0.0907

All electrical power and voltages are shown in PER UNIT and the rotor angles are in rad.

Table B-8 Load parameters for 2nd operating point

L_1	L_2	L_3
3.755 - j2.5	4.25 - j2.5	3.5 - j2.25

All resistances and reactances are shown in PER UNIT.

Appendix C - QR Factorization

The QR factorization of an *m*-by-*n* matrix A is given by [47][57]

$$A = Q \cdot R \tag{C.1}$$

where $Q \in \Re^{m \times m}$ is orthogonal, that is $Q \cdot Q^T = I$ and $R \in \Re^{m \times n}$ is upper triangular.

One implementation of the QR factorization is to solve the linear equation

$$\Phi \cdot x = b \tag{C.2}$$

for $x \in \Re^{n \times 1}$, where $\Phi \in \Re^{n \times n}$ and $b \in \Re^{n \times 1}$ are known. After QR factorization of the matrix $\Phi \rightarrow Q \cdot R$ and using the orthogonal property of Q matrix,

$$R \cdot x = Q^T \cdot b \tag{C.3}$$

the equation can be solved by back-substitution.

The standard algorithms for QR factorization involve successive Householder reflection or Givens rotation.

An appropriate Householder matrix applied to a given matrix can zero all elements in a column of a matrix situated below a chosen element. Thus, the first Householder matrix, H_1, can be selected to zero all elements in the first column of matrix A, below the first element. Similarly, matrix H_2 zeros all elements in the second column below the second element, and so on, up to H_{n-1}. Consequently,

$$R = H_{n-1} \cdot \ldots \cdot H_2 \cdot H_1 \cdot A . \tag{C.4}$$

Since the Householder matrices are orthogonal

$$Q = (H_{n-1} \cdot \ldots \cdot H_2 \cdot H_1)^{-1} = H_{n-1} \cdot \ldots \cdot H_2 \cdot H_1 . \tag{C.5}$$

The algorithm for the Givens rotation is very similar, except that it only introduces one zero at a time, so it requires $t = (n-1)^2/2$ rotations

$$Q = G_1 \cdot G_2 \cdot \ldots \cdot G_t . \tag{C.6}$$

The nomenclature QR is originating from the fact that for an upper triangular matrix Q the *right* of the matrix is non zero. In a same way the QL algorithm will produce a lower triangular matrix L where the *left* of the matrix is non zero.

Appendix D - The Lyapunov Equation

Let the plant dynamic be defined with the state and output equations [54]

$$x_{n+1} = A_n \cdot x_n + B_n \cdot u_n \tag{D.1}$$

$$y_n = C_n \cdot x_n \tag{D.2}$$

where $x_n \in \Re^m$, $u_n \in \Re^r$, $y_n \in \Re^q$ and the associated matrices have the appropriate dimensions. If the plant is uncontrolled, $u_n = 0, \forall n \in (0, N)$, the general quadratic performance index of the form

$$J = x_N^T \cdot P_N \cdot x_N + \sum_{n=1}^{N-1} \begin{bmatrix} x_n^T & u_n^T \end{bmatrix} \cdot \begin{bmatrix} Q_n & S_n \\ S_n^T & R_n \end{bmatrix} \cdot \begin{bmatrix} x_n \\ u_n \end{bmatrix} \tag{D.3}$$

will be reduced to

$$J = x_N^T \cdot P_N \cdot x_N + \sum_{n=1}^{N-1} x_n^T \cdot Q_n \cdot x_n \ . \tag{D.4}$$

In order to determine the behavior of the plant in uncontrolled environment, the cost J_n should be evaluated in the interval $\{0, \ldots, n, \ldots, N\}$. This evaluation can be done using backward recursion. The value of the cost in the final time interval is defined as

$$J_N = x_N^T \cdot P_N \cdot x_N \ . \tag{D.5}$$

Now, let $n = N - 1$ and the cost can be expressed as

$$J_{N-1} = x_N^T \cdot P_N \cdot x_N + x_{N-1}^T \cdot Q_{N-1} \cdot x_{N-1} \quad . \tag{D.6}$$

Using the plant dynamics (D.1) for the uncontrolled plant $x_N = A_{N-1} \cdot x_{N-1}$, the cost can be written as

$$J_{N-1} = x_{N-1}^T \cdot (A_{N-1}^T \cdot P_N \cdot A_{N-1} + Q_{N-1}) \cdot x_{N-1} \quad . \tag{D.7}$$

The content in the brackets can be defined as a backward recursion for a new variable P_n,

$$P_{N-1} = A_{N-1}^T \cdot P_N \cdot A_{N-1} + Q_{N-1} \quad . \tag{D.8}$$

With the backward recursion variable P_n, the cost (D.6) is

$$J_{N-1} = x_{N-1}^T \cdot P_{N-1} \cdot x_{N-1} \tag{D.9}$$

which has the same form as the cost for the final state (D.5). According to the recursion (D.9), all costs can be calculated. The backward recursion for this intermediate variable P_n is

$$P_n = A_n^T \cdot P_{n+1} \cdot A_n + Q_n \qquad n = \{N-1, \ldots, n, \ldots, 0\} \quad . \tag{D.10}$$

This is the Lyapunov recursive equation, which calculates the performance index kernel sequence P_n. The Lyapunov recursive equation can be solved off-line, using only the system state transformation matrix A_n. The performance index kernel sequence, P_n, can be used to calculate the cost for the uncontrolled plant for any initial state x_n.

If the weighting matrix $Q_n > 0$ (positive definite), then P_n will be positive definite as well ($P_n > 0$).

Lyapunov theory. If the plant is internally stable and the selected $Q > 0$, then the Lyapunov equation has a positive definite solution, $P > 0$.

In the time invariant case $A_n = A$ the recursion can be written as

$$P_n = (A^T)^{N-n} \cdot P_N \cdot A^{N-n} + \sum_{k=n}^{N-1} (A^T)^{N-k-1} \cdot Q \cdot A^{N-k-1} \tag{D.11}$$

as $(N-n) \to \infty$, (D.11) converges to steady state value

$$P_\infty = \sum_{k=0}^{\infty} (A^T)^k \cdot Q \cdot A^k \tag{D.12}$$

and as $P = P_n = P_{n+1}$, it become the algebraic Lyapunov equation

$$P = A^T \cdot P \cdot A + Q. \tag{D.13}$$

Second norm of the observations theorem. [7] Consider the signal generated by the system (D.1) and (D.2) with $u_n = 0, \forall n \in (0, N)$ in which every entry of A_n and C_n is bounded and $y_n \in l_2[0, N]$ for all x_0. Then, the second norm of the observations y_n, on the interval $[0, N]$, can be calculated as

$$\|y\|^2_{2,[0,N]} = x_0^T \cdot P_0 \cdot x_0 \tag{D.14}$$

where P_0 is obtained by the backward Lyapunov recursion $n = N, N-1, \ldots, 0$

$$P_n = A_n^T \cdot P_{n+1} \cdot A_n + C_n^T \cdot C_n \qquad P_{N+1} = 0 \ . \tag{D.15}$$

***l_2* space definition.** The function f_n belongs to the l_2 space if it has m observable outputs where each sequence of observations is in $\Re^m$, defined by

$$l_2[0, N] = \{f \ : \ f_n = 0 \ \text{ for } \ n \notin [0, N], \ \|f_n\|_{2, [0, N]} < \infty\} . \tag{D.16}$$

The second norm is calculated by

$$\|f_n\|_{2, [0, N]} = \left\{ \sum_{n = 0}^{N} f_n^T \cdot f_n \right\}^{1/2} . \tag{D.17}$$

For the function f_n to have the properties defined above, the shorthand $f_n \in l_2[0, N]$ is often used.

Appendix E - Discrete-Time System Theory

E.1 Stability of Discrete-Time Systems

The discrete-time system [7]

$$x_{n+1} = f(n, x_n, u_n) \tag{E.1}$$

$$y_n = h(n, x_n) \tag{E.2}$$

is stable if $\{y_n;\ n \in [0, N]\}$ is in l_2 space, whenever the input sequence is $\{u_n;\ n \in [0, N]\}$ in l_2 space.

Stability theorem. The following conditions are equivalent

1. The closed-loop system, without external effects,

$$\left[x_{n+1}\right] = A \cdot \left[x_n\right] \tag{E.3}$$

is internally stable if, from any finite initial state, it will converge to zero final state as $n \to \infty$.

2. If $\rho(A) < 1$, all eigenvalues of A have magnitudes less than 1.

3. For any positive symmetric matrix Q there exists a unique positive definite symmetric matrix P, which is the solution of the following algebraic Lyapunov equation

$$P = A^T \cdot P \cdot A + Q. \tag{E.4}$$

E.2 Stabilizability of Discrete-Time Systems

A system is stabilizable if, and only if, every unstable mode is controllable.

Stabilizability theorem. The following statements are equivalent [80]

1. (A, B) is stabilizable.
2. The matrix $\left[A - \lambda \cdot I \; B\right]$ has full row rank for all $\lambda \geq 1$.
3. There exists a matrix F, such that $\left[A + B \cdot F\right]$ is stable.

E.3 Controllability of Discrete-Time Systems

The discrete-time system

$$x_{n+1} = f(n, x_n, u_n) \tag{E.5}$$

is controllable/reachable if, for any initial state x_0, there exists a finite control sequence $\{u_n;\ n \in [0, N]\}$ that will transfer the state x_0 to x_N, where x_N is any selectable finite state.

Controllability theorem. The following statements are equivalent [80]

1. The linear time invariant system of m^{th} order

$$x_{n+1} = A \cdot x_n + B \cdot u_n, \tag{E.6}$$

where $x_n, x_{n+1} \in \Re^m$, $u_n \in \Re^r$, is controllable/reachable.

2. The controllability gramian

$$P_n^C = \sum_{k=0}^{n} A^k \cdot B \cdot B^T \cdot (A^T)^k \tag{E.7}$$

is nonsingular for some $n < \infty$.

3. The controllability matrix

$$W^C = \left[B \; AB \; A^2B \; \ldots \; A^{m-1}B \right] \tag{E.8}$$

has rank *m.*

4. The eigenvalues of $[A + B \cdot F]$ can be freely assigned (with the restriction that complex eigenvalues are in conjugate pairs) by a suitable choice of F.

5. The matrix $\left[A - \lambda \cdot I \; B \right]$ has full row rank for all λ in $\Im$.

E.4 Observability of Discrete-Time Systems

The discrete-time system

$$x_{n+1} = f(n, x_n, u_n) \tag{E.9}$$

$$y_n = h(n, x_n) \tag{E.10}$$

is observable if for any initial state x_0 there exist a finite number of observations of the output sequence $\{y_n;\ n \in [0, N]\}$ and the knowledge of the input sequence $\{u_n;\ n \in [0, N]\}$ is such that the initial state x_0 can be determined.

Observability theorem. The following statements are equivalent [80]

1. The linear time invariant system of m^{th} order

$$x_{n+1} = A \cdot x_n + B \cdot u_n \tag{E.11}$$

$$y_n = C \cdot x_n \tag{E.12}$$

   where $x_n, x_{n+1} \in \Re^m$, $u_n \in \Re^r$, and $y_n \in \Re^q$ is observable.

2. The observability gramian

$$P_n^O = \sum_{k=0}^{n} (A^T)^k \cdot C^T \cdot C \cdot (A)^k \tag{E.13}$$

   is nonsingular for some $n < \infty$.

3. The observability matrix

$$W^O = \begin{bmatrix} C \\ CA \\ \cdots \\ CA^{m-1} \end{bmatrix} \tag{E.14}$$

   has rank *m.*

4. The eigenvalues of $[A + L \cdot C]$ can be freely assigned (with the restriction that complex eigenvalues are in conjugate pairs) by a suitable choice of L.

5. The matrix $\begin{bmatrix} A - \lambda \cdot I \\ C \end{bmatrix}$ has full column rank for all λ in $\Im$

E.5 Detectability of Discrete-Time Systems

A system is detectable if, and only if, every unstable mode is observable.

Detectability theorem. The following statements are equivalent [80]

1. (C,A) is detectable.

2. The matrix $\begin{bmatrix} A - \lambda \cdot I \\ C \end{bmatrix}$ has full column rank for all $\lambda \geq 1$.

3. There exists a matrix L, such that $\left[A + L \cdot C\right]$ is stable.

Appendix F - Test for Positive Definiteness

The positive definite test is an essential part of the H_∞ algorithm. A matrix $A \in \Re^{m \times m}$ is positive definite if $x^T \cdot A \cdot x > 0$ for all nonzero $x \in \Re^m$. The algorithm presented here is a compromised solution between the required computational power and memory. The positive definite test is based on the property of a positive definite matrix $A > 0$ which, when factorized as $A = L \cdot D \cdot M^T$, the diagonal matrix $D = \text{diag}(d_1, d_2, \ldots, d_k)$ has positive diagonal elements, where L is a *unit lower triangular* and M^* is a *unit upper triangular* [57].

```
 x is an [m x m] Matrix

ptest(Matrix x)  {
  % return ptest = TRUE if  'x' is a Positive definite matrix (ptest=FALSE otherwise)
    flim = 1e-9;
    m=x->rows;
    for(j=1; j<=m; j++)
        {
             for(i=1; i<=j; i++){
                       sum1=0;
                        for(k=1; k<i; k++) sum1 += x[i][k] * x[k][j];
                        x[i][j] = x[i][j] - sum1;            }
              if(x[j][j] < flim)   {  return  FALSE,    j=m;    }
              else   {
                        for(i=j+1; i<=m; i++){
                                  sum2 = 0;
                                  for(k=1; k<j; k++) sum2 += x[i][k] * x[k][j];
                                   x[i][j] = (x[i][j] - sum2)/x[j][j];      }    }
         }
    return TRUE;     }
```

Appendix G - ARMA Model Stability Test

The Schur-Cohn stability test is the backward Levinson recursion, when used to determine filter stability. All of the reflection coefficients of an AR process must be less than one in magnitude for the filter to be stable. This can easily be tested by the algorithm presented below [47].

θ <=> [DC, a_1, a_2, a_3,...a_{m_a}, b_1,...] if m_s = 1
where: ϕ = [1,-y_{n-1},...-y_{n-m_a},u_{n-1},...u_{n-m_b},v_{n-1},...v_{n-m_c}]

θ <=> [a_1, a_2, a_3,...a_{m_a}, b_1,...] if m_s = 0
where: ϕ = [-y_{n-1},...-y_{n-m_a},u_{n-1},...u_{n-m_b},v_{n-1},...v_{n-m_c}]

Backward Levinson (step-down) recursion:

```
stable (Vector θ) {
    % return Stable=TRUE if  'θ' is a stable ARMA model (Stable=FALSE otherwise)
     f = θ[m_s+1];
     for(i=1;i < m_a;i++)   Vb[i] = Va[i] = (θ[m_s+i+1] /  f) end
     f = Va[m_a-1];          //     Γ(1)
     if( abs(f) > 1 ) Stable=FALSE,  return;
      for(j=m_a-1;j>=2;j--)
          {
                for(i=1;i<=j;i++)     Va[i] = Va[i] - (f * Vb[j-i]);
                 f = 1 - (f * f );
                for(i=1;i<=j-1;i++)   Va[i] = Vb[i] = Va[i]/f  ;
                f = Va[j-1];                    //   Γ(i)
               if( abs(f) > 1) return FALSE ;
            }
     return TRUE ;      }  // system model is stable all  |Γ(i)| < 1
```

Index

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

V

W

Z