BOOTSTRAP ESTIMATION OF CONTROLLER PERFORMANCE INDICES AND DETECTION OF MODEL MISMATCH

By

Jonathan R. Webber

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering

at

DALHOUSIE UNIVERSITY

Halifax, Nova Scotia September 2006

Your file Votre référence
ISBN: 978-0-494-27652-5
Our file Notre référence
ISBN: 978-0-494-27652-5

DALHOUSIE UNIVERSITY

To comply with the Canadian Privacy Act the National Library of Canada has requested that the following pages be removed from this copy of the thesis:

Preliminary Pages
- Examiners Signature Page
- Dalhousie Library Copyright Agreement

Appendices
- Copyright Releases (if applicable)

Dedication page

I dedicate this thesis to my family, for their support and frequent dinners, and for eventually learning that asking "when will I finish?" does not help accelerate the process. I also dedicate this thesis to Orasa, who never got tired of me saying that I was too tired from all my hard work and didn't want to talk on the phone. And to all my friends who understood why I couldn't come out and play because I had 'one more' simulation to complete.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Acronyms:

AIC :	Akaike Information Criterion
AR:	autoregressive model
ARIMA:	autoregressive integrated moving average model
ARMA:	autoregressive moving average model
BC:	bias corrected percentile method
DMC:	dynamic matrix control
FCOR:	filtering and correlation method
FIR:	finite impulse response
FOPDT:	first order plus dead-time model
IDCOM:	identification and command control
iid:	independent and identically distributed
IMC:	internal model control
MBC:	model based control
MPC:	model predictive control
MIMO:	multiple input/multiple output
MPM:	model plant mismatch
PID:	proportional, integral, and derivative controller
QDMC:	quadratic dynamic matrix control
se:	standard error
SISO:	single input/single output
SMPC:	simplified model predictive control
VAR:	vector autoregressive model

Roman:

A:	dynamic matrix
a_i	innovations (random sequence)

$\hat{a}$:	estimated innovations
B:	number of bootstrap replications
b:	VAR lag parameter
C:	covariance
$\hat{c}$:	estimated covariance
D:	process delay, $D \geq 1$
$\hat{D}$:	estimated process delay
E:	expectation operator
e:	error to be compensated for DMC
F:	population distribution
$\hat{F}$:	empirical distribution
G:	process transfer function
$\hat{G}$:	process model
G^*:	process transfer function, delay free
G_d:	disturbance transfer function
g_i:	impulse response coefficient
$\hat{g}_i$:	estimated impulse response coefficient
H:	step response coefficient
K_G:	gain of transfer function G
N:	sample size
N_b:	number of 'burn in' samples
N_p:	prediction horizon
N_u:	number of inputs
N_y:	number of outputs
N_s:	settling time
N_Υ:	dimension of vector Υ
$N(0,1)$:	normal distribution, with mean zero and unit variance
M:	control move horizon
P_α:	α^{th} percentile
Q:	IMC controller

r:	set-point
R:	cross correlation
S:	sum of squared errors
T:	generic statistic
t:	time
u:	process input
u_d:	input dithering
y:	process output
$\hat{y}$:	predicted process output
z:	discrete shift operator, or normal variate.

Greek:

A:	VAR coefficient matrix
α:	coefficients of A matrix
ε:	prediction error
λ:	IMC tuning parameter
Φ:	vector of time series parameters, or population parameter
$\hat{\Phi}$:	estimated population parameter
$\hat{\Phi}^*$:	bootstrap estimated population parameter
ρ:	auto correlation
θ:	model mismatch vector
σ^2:	variance
Π:	closed loop transfer function between ε and r
Σ:	covariance matrix
τ:	time constant
τ_s:	closed loop settling time
Υ:	time series observations
υ:	disturbance

Γ:	cumulative standard normal distribution
ξ:	time series model
ψ:	various closed loop transfer functions
ψ_i:	i^{th} closed loop impulse response coefficient
Ω:	closed-loop transfer function between u and r
ζ:	minimum variance performance index
$\hat{\zeta}$:	estimated minimum variance performance index

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Gupta for his encouragement not to quit when things seemed impossible, and also for his unlimited patience and friendship. I thank Dr. Kuzak, and Dr. Sandblom for acting on my committee and also for their friendship, encouragement and support during my studies. Finally, I thank Dr. Gupta, NSERC, Killam trusts and Dalhousie for their financial support.

I would also like to thank Dr. Perrier for serving as my external examiner, and finding my oversights. It was certainly an honor and a pleasure to have Dr. Perrier attend my defense presentation.

ABSTRACT

Since controllers play a vital role in industrial processes, it is important to have methods that monitor controller performance and diagnose the reasons when performance is poor. This thesis deals with the inherent uncertainty involved in measuring the controller performance. Since a common cause of poor performance in model-based controllers is model error, this thesis also proposes several closed-loop methods for diagnosing the presence of model error.

The estimates of the statistical properties of the minimum variance controller performance index need to be made by using small samples since in practice the process data may only be stationary over short intervals. Residual based bootstrapping is used to estimate these statistical properties. It is demonstrated that accurate confidence intervals can be obtained by using small samples in case of normal and non-normal innovations. The broader applicability of the bootstrapping approach is also demonstrated by estimating the sampling distribution of the closed loop settling time performance index. An experimentally determined sampling distribution for the minimum variance performance index shows that the index can have strong non-normal behavior. The distribution estimated by the proposed bootstrap method is shown to capture the overall features of the experimental distribution. Also, a bootstrap method is proposed which can capture the effects of model-order uncertainty on the sampling distribution of the performance index.

While bootstrapping of the controller performance index addresses the detection of performance issues, several closed loop methods are also proposed to diagnose (and in some cases correct) model-plant mismatch. An iterative method is proposed for univariate systems to detect and correct gain and dead-time mismatch for models of arbitrary order. This method is also used to correct for time constant mismatch in first order plus dead time models. A multivariate cross-correlation method is presented to help detect which specific models in model-based controllers are mismatched. A partial control method is developed which utilizes the patterns contained in the closed loop step response of the prediction error to set point changes to assist in the determining which models are mismatched in multivariate model-based controllers. A sieve bootstrap method is developed to estimate the confidence bands for the impulse response functions determined via closed loop correlation analysis. Further the method is used to estimate the confidence intervals for the process gains. This method can help detect the presence of statistically significant model mismatch, and also the extent of the mismatch.

CHAPTER 1 INTRODUCTION AND CONTRIBUTIONS

1.1 Introduction

Automatic process control is vital for the safe and economic operation of today's complex industrial processes. Controllers must regulate unpredictable disturbances, and at times move the process variables to new set-points. Control is achieved by using a sensor to measure the controlled variable and subtracting the measured value from the set-point. This subtraction results in an error-term, which is then used as an input to the controller. Based on the error, the controller calculates a process input, which is then applied to the system. The most common industrial controllers are based on proportional, integral and derivative control actions. Since the 1970s, control algorithms which explicitly use a process model (model based controllers) have become the most widely used "advanced" form of control in industry. Since controllers play a vital role in industrial processes, it is important to have methods that monitor controller performance and diagnose the reasons when performance is poor. This thesis deals with the inherent uncertainty involved in measuring the controller performance by using bootstrap statistical techniques. Since a common cause of poor performance in model-based controllers is model error, this thesis also proposes several closed-loop methods for diagnosing the presence of model error.

Controller performance monitoring involves the comparison of actual controller performance to a relevant benchmark or design objective. This allows the engineer to determine if the controllers are performing as designed or to monitor for changes in their performance. Estimators of controller performance are functions of the process output, and therefore are random variables due to stochastic disturbances. To make sound statistical inferences about performance measures, it is important to know how these estimators are distributed. The sampling distributions of these estimates allow the determination of confidence intervals and quantitatively describe the uncertainty of the performance measure.

If the controller performance is found to deviate from acceptable performance (by using the confidence intervals described above), then the root cause of poor performance must be diagnosed. Diagnosing the root cause of poor controller performance is an important yet difficult task due to the many possible contributing factors (e.g. sensor failures, poor tuning). For model based controllers, the poor performance is usually due to model-plant mismatch (MPM). MPM refers to differences between the actual process and model, and can degrade the performance of a control system and/or cause instability. The model mismatch may arise due to poor initial identification, or result from changing process parameters (e.g. fouling of a catalyst). The presence of MPM directly affects controller performance because the controller must be detuned (i.e. tuned for sluggish performance) if it is to remain stable in the presence of model error. If MPM is detected, the control engineer can decide if it is worthwhile to re-identify the inaccurate process model, and subsequently redesign a more aggressive controller. Therefore, this thesis also focuses on the detection (and in some cases the correction) of model-plant mismatch (MPM) in model-based controllers.

1.2 Background and contributions

The following sections provide a summary of previous work, and a description of the contributions introduced in this thesis.

1.2.1 Sampling distributions of performance indices

By far, the most popular performance measure for controllers regulating stochastic disturbances has been the minimum variance benchmark, which compares the actual process variance to the lowest achievable [1]. Existing methods for estimating the variance or developing confidence intervals for the minimum variance index have been based on asymptotic arguments and parametric assumptions concerning the distribution of the innovations (innovations are the independently distributed random variables

exciting the system, and represent the unpredictable portion of the process output) [2-4]. Methods that provide variance estimates further assume that the estimator is normally distributed when determining confidence intervals. However, simulations and experimental results provided in this thesis indicate that the performance index can deviate significantly from normality. In addition, the existing methods have not attempted to estimate the bias of the performance index, or how to estimate the distribution of other performance indices such as the closed-loop settling time.

Bootstrapping was introduced by Efron in 1979 as a flexible method for estimating the sampling distribution of a statistic, which is an estimator for some population parameter [5].While bootstrapping has been widely used in the statistics and econometrics literature, it is a relatively unknown technique in chemical engineering. Since bootstrapping does not rely on asymptotic distributions, or parametric assumptions concerning the innovations, it has the potential to perform better than existing methods, especially when the underlying assumptions of the previous methods are invalid. Previous works have been simulation based, and apparently no real-wrold sampling distributions of the performance index have been presented. This thesis presents an experimentally determined sampling distribution of the performance index for a bench scale liquid level controller.

Also, the previous methods have considered that the model order is fixed. That is, over the different realizations of the process output, the same order time-series model is used each time. This appears unrealistic, since the user will not have a preordained model order. In practice some criteria such as the Akaike Information Criterion (AIC) are used to select the model order. Therefore, the model order will also be a random variable. It is demonstrated that the model order selection uncertainty can have a large impact on the variance of the performance index. It is not clear how the previous methods can be adapted to handle such considerations. However, it is shown in this thesis that the bootstrap framework can naturally accommodate the model order selection uncertainty.

CONTRIBUTIONS/FINDINGS

- Residual based bootstrapping is used to estimate the bias, variance, and confidence intervals of the minimum variance performance index.

- The bootstrap method is compared to previous methods for estimating confidence intervals for the minimum variance index. It is demonstrated that for small sample sizes and/or strongly non-normal innovations the bootstrap method can provide better performance. Recommendations are made as to when each method is appropriate.

- It was found that bootstrapping estimates of the bias are not reliable. Bias estimates appeared to be unbiased, but had a large variance, which would inflate the variance of the performance index.

- Bootstrapping is used to estimate the variance of the closed loop settling time index, for which there was no previous method to estimate its uncertainty.

- An experimentally determined sampling distribution for the minimum variance performance index is obtained from a bench scale liquid level controller. For smaller sample sizes, strongly non-normal distributions are observed. The distribution estimated using the bootstrap is shown to capture the overall features of the experimental distribution.

- It is demonstrated that the model order uncertainty can have a significant effect on the uncertainty of the performance index.

- Bootstrapping is used to model the effect of model order uncertainty on the sampling distribution of the performance index.

1.2.2 Closed loop detection of model plant mismatch

If we open the control loop, it is relatively easy to determine if the models are mismatched. However, for safety, and economic reasons it is usually undesirable to open the control loop. Therefore this thesis considers several closed-loop methods for detecting model mismatch. Since the most popular form of model predictive control (dynamic matrix control, DMC) utilizes a simple disturbance model, the methods used in this thesis are insensitive to disturbance model mismatch and detect only process model mismatch. Many of the methods in the literature provide yes/no (e.g. [6-12]) answers to the presence of MPM; however, this thesis also considers the detection of specific parameter mismatches such as gain and dead-time mismatch. Moreover, there has been little work in the area of identifying which subset of multivariable models should be considered for re-identification; therefore this thesis considers methods to aid in the selection of candidate models for re-identification.

A method developed by Webber et al. [13] used the patterns in the prediction error resulting from step changes in the set-point to assess the presence of specific parameter mismatch (i.e. gain, time-constant, and dead-time). The method considered first-order plus dead-time models (FOPDT) used to design controllers for FOPDT processes. However, as demonstrated in this thesis, it can be difficult to distinguish between time constant and dead-time mismatch using the previous method. This thesis extends these ideas to develop iterative methods for detecting and correcting gain and dead-time mismatch for models of any order. In the case of FOPDT models, after correcting the dead-time, any ambiguity between dead-time and time-constant mismatch can be removed. This allows an iterative method to be developed to correct all the parameters of FOPDT models based on the pattern of the prediction error. This method is also demonstrated to be effective when the process is not FOPDT, but instead a higher order process.

In practice, the identification of models is a very time consuming process, and becomes even more complicated for multiple-input/multiple-output (MIMO) systems. If

model mismatch is detected, it can be valuable to determine which specific input-output pairings contain the mismatch. For example a 10 x 10 MIMO system would have 100 individual models. If the subset of mismatched models can be identified, then the re-identification process (either open or closed-loop) can focus only on the subset of models that require re-identification. This can save time, and simplify the re-identification process. Also, by identifying fewer parameters the variance of the estimated parameters can be reduced [14]. In [6] cross-correlations were used to help detect the presence of model mismatch for SISO (single input-single output) control systems. This method is extended in this thesis to MIMO (multiple input-multiple output) systems to help detect which rows and columns of the transfer function matrix contain mismatch. Webber *et* al. [13] detected which specific input-output models are mismatched in MIMO control systems by holding a single input constant during the set-point change. This thesis demonstrates that it may be inadequate to only hold a single input constant, and that it may be required to hold various subsets of the input constant. The previous method also used transient analysis which is sensitive to disturbances and requires large set-point deviations to observe the pattern which may excite process non-linearities. This thesis improves the situation by considering correlation analysis which requires smaller amplitude deviations in the set-point and is more 'robust' to disturbances.

The need to detect MPM naturally leads to the consideration of system identification techniques. However, parametric modeling can be difficult, especially in multivariable systems. Therefore, non-parametric methods are better suited for screening for the presence of MPM. The two most common forms of nonparametric methods are spectral and correlation approaches. This thesis considers the correlation approach. Correlation analysis is a standard system identification method for directly estimating the impulse response function [14-16] . It is shown that if the non-prewhitening approach is used, a significant reduction in the variance of the estimated coefficients can be obtained. However, it is more difficult to analytically determine the variance when the inputs are not considered as perfect white noise. Therefore a sieve bootstrapping method is developed to estimate the confidence bands for impulse response functions and confidence limits for the process gain determined via correlation analysis. Since DMC and IDCOM uses step/impulse response models, it can be useful to directly calculate the

mismatch in the time domain. Further, some robust model predictive control schemes are developed which explicitly use time domain uncertainty [17]. The results are then extended to MIMO systems, and ultimately to closed-loop MIMO control systems. In the literature closed loop identification via correlation analysis has received little attention. Using the methods presented in this thesis, uncertainty bands can be used to explicitly determine the presence of statistically significant model error, and also the magnitude of the error.

CONTRIBUTIONS/FINDINGS

- The transient response of the prediction error to set-point changes is used for univariate systems to detect and correct gain and dead-time mismatch for models of any order. An iterative procedure is presented.
- The above procedure is then used to correct for gain, dead-time, and time-constant mismatch in model based controllers designed around a FOPDT model. It is demonstrated that the method can be effective in correcting specific parameter mismatch even when the process is a higher order system.
- The cross-correlation method presented in [6] for univariate systems is extended to MIMO systems to help detect which specific input-output pairing contains model mismatch.
- The partial control method of [13]is improved by considering correlation analysis instead of transient analysis. This allows smaller set-point changes to be used and decreases sensitivity to disturbances.
- It is demonstrated that using a non-prewhitening method for correlation analysis can reduce the variance of the estimated impulse response coefficients.
- A sieve bootstrap method is used to estimate the confidence bands for the impulse response functions for both SISO and MIMO systems. Further the method is used to estimate the confidence intervals for the process gains.

- The sieve bootstrap method is further extended to closed-loop identification of model error for MIMO control systems. It can help detect the presence of statistically significant model mismatch, and also the extent of the mismatch.

1.3 Thesis outline

Chapter 2: Provides an overview of the statistical performance monitoring literature. The bootstrap method is introduced and is used for estimating the sampling distribution of performance measures

Chapter 3: First provides an overview of MPC. Then various closed-loop methods for detecting the presence of model mismatch for SISO and MIMO control systems are presented.

Chapter 4: A sieve bootstrap method is presented for SISO and MIMO correlation analysis to estimate the uncertainty of estimated impulse response coefficients and confidence limits for the process gains. The method is then extended to closed-loop identification of the model mismatch.

CHAPTER 2 SAMPLING DISTRIBUTION OF PERFORMANCE INDICES

2.1 Statistical controller performance assessment

This section provides an overview of statistical performance monitoring literature. Controller performance assessment has become an established field in control research, with several review articles, and even a textbook describing its development [10, 18-22]. The seminal paper in controller performance monitoring by Harris [1] showed that normal operating data can be used to estimate the minimum achievable output variance for single-input/single-output (SISO) systems provided that the process delay is known. The work of Harris extended the work of [23], who used multivariate time series analysis to evaluate the performance of controllers in the paper industry but did not consider process delays. Since process delays are common in industry, and strongly limit the achievable performance this was an important extension.

The minimum achievable variance is a natural benchmark for controllers regulating stochastic disturbances. A performance measure is simply the ratio of the actual variance to the best achievable variance, or some function of the ratio. While many controllers are not actually implemented as minimum variance controllers, the minimum achievable variance still provides an absolute lower bound on the performance. The lowest achievable variance is limited by the disturbance characteristics and the process delay. Thus no amount of controller redesign can improve the situation if the desired output variance is lower than the minimum achievable variance. In this case only reduction of the process delay or reduction of the disturbance by process redesign or the addition of feed-forward control can help.

The original paper by Harris utilized an autoregressive moving average (ARMA) time series model [1]. Since the moving average part of the model can cause some difficulties in the identification, Desborough and Harris have suggested the use of an autoregressive model (AR) to determine a normalized performance index [24]. This method is sometimes referred to as the R^2 method. Laguerre models have also been considered [25]. More recently ARMarkov models have been considered [26]. An alternative approach that bypasses the calculation of the impulse response coefficients is the filtering and correlation method (FCOR) introduced by Huang and Shah [22]. Further, it has been known for some time that the autocovariance function can be used to check if the control loop is operating as a minimum variance controller without any further modeling [27]. However, this approach only provides a yes/no answer regarding the achievement of minimum variance.

After the initial developments in assessing feedback systems, attention was turned to assessing feedback/feed-forward control systems. An analysis of variance has been used to determine the potential reduction in the output variance that can be achieved by incorporating a feed-forward controller for a given measured disturbance [24]. The performance of feed-forward controllers was also examined using cross-correlation between measured disturbance and the controlled variable [6]. The performance assessment of feed-forward controllers has also been extended to MIMO systems [28].

Naturally, other extensions of the original idea followed. A frequency domain approach was used to examine the performance of controllers by examining the bandwidth and maximum magnitudes of the sensitivity functions [8, 9], which tied the assessment to robust control theory. Another interesting extension was the consideration of the best achievable performance with structural constraints on the controller. Along these lines the best achievable PI or PID benchmarks were introduced [29-31]. Many common performance measures (such as closed-loop settling time and decay rate) have been formulated in terms of the closed-loop impulse response coefficients [32]. Assuming normally distributed innovations, they used generalized likelihood tests to

determine if the closed-loop satisfied some performance objective. Probably the most important extension after the seminal work by Harris was the extension of the minimum variance benchmark to multivariate systems [33, 34]. This required the introduction of a process interaction matrix, which can be interpreted as a sort of multivariate delay. Generally, the interactor matrix cannot be determined from knowledge of the process delays alone. A two-step procedure was proposed in [33], which consisted of a multivariate spectral factorization, and a solution of a multivariate Diophantine equation to estimate the minimum variance, assuming the interactor matrix is known. Alternative methods based on knowledge of the first few Markov matrices have also been proposed [34, 35]. Assuming the disturbance model is known for MIMO systems, the closed loop impulse response coefficients are compared to user designed coefficients [36]. Instead of placing all closed loop poles at the origin (i.e. minimum variance), a benchmark with one pole located at a user defined location has been considered [3]. The performance assessment of time varying systems, and time variant disturbances has been studied in [37] [38] [39]. The effects of model mismatch on performance were considered in [11, 40]. A projection based performance monitoring approach is adopted in [41, 42]. The effect of process nonlinearity on the performance has been studied in [43].

Recently there has also been interest in finding suitable performance measures for model predictive controllers. Hugo points out that the minimum variance benchmark does not consider the limiting effects of the structure of dynamic matrix control (DMC) (e.g. simplified disturbance model) [44]. He suggests using a performance index, which is based on a random walk disturbance model, as used in DMC instead of a full autoregressive integrated moving average (ARIMA) model. The standard minimum variance performance measure does not account for penalties on the controller input variance, thus the linear quadratic performance index was proposed, which requires an accurate process and disturbance models [22]. A normalized multivariate impulse response curve, which is a lumped measure and does not require knowledge of the interactor matrix has been suggested in [45]. The ratio of designed to achieved objective function can be used to assess the performance of constrained model predictive

controllers, but does not provide information concerning the best achievable performance [46]. A historical benchmark which compares the current performance to when the performance was known to be good has also been proposed [46]. The optimal objective function for move and prediction horizons are suggested as a benchmark [47], where they have integrated monitoring and some diagnostics. The performance monitoring of SISO model predictive controllers using a performance index that accounts for the step disturbance model commonly used in practice has been considered in [48]. This method requires knowledge of the true process and disturbance dynamics. The expected performance is used as a benchmark for input constrained model predictive controllers [49].A constrained minimum variance controller based on a moving horizon approach is used to assess model predictive controllers, which also accounts for constraints on the process variables [50].

2.1.1 Calculating the SISO benchmark

Several different benchmarks may be used to monitor controller performance. I have considered the minimum variance and the closed-loop settling time as indicators of performance. The settling time is considered to demonstrate the generality of the bootstrap method. This section briefly outlines how point estimates of these performance measures might be obtained. Further details may be found in the previously cited reviews on performance assessment.

The developments in this chapter are based upon the following linear time invariant system:

$$y_t = G^*(z^{-1})z^{-D}u_t + G_d(z^{-1})a_t \tag{2.1}$$

where $G^*(z^{-1})$ and $G_d(z^{-1})$ are the process and disturbance transfer functions, respectively. Note that $G^*(z^{-1})$ is the delay free transfer function. *D-1* represents the process delay ($D \geq 1$), and a_t is a zero mean, independent and identically distributed (*i.i.d.*) sequence with variance σ_a^2. The process input, u_t, is determined by some linear feedback control law. The closed-loop transfer function between y_t and a_t may be expressed as:

$$y_t = \psi(z^{-1})a_t = (\psi_0 + \psi_1 z^{-1} + \ldots + \psi_{D-1} z^{-D+1})a_t + \sum_{i=D}^{\infty} \psi_i z^{-i} a_t \tag{2.2}$$

where the ψ_i for $i = 0,1,\ldots D-1$ are controller invariant. If the system is operating under minimum variance control then $\psi_i = 0 \ \forall i = D,\ldots,\infty$. Hence, the minimum output variance may be calculated from Eq.(2.2):

$$\min Var(y_t) = \sigma_{\min}^2 = (\psi_0^2 + \psi_1^2 + \ldots + \psi_{D-1}^2)\sigma_a^2 \tag{2.3}$$

A typical performance index is the ratio of actual output variance σ_y^2 to $\sigma_{\min}^2$ [19]:

$$\xi = \frac{\sigma_y^2}{\sigma_{\min}^2} \tag{2.4}$$

An estimate for the minimum variance may be obtained by fitting a time-series model to the output y_t and substituting the estimated impulse response coefficients $\hat{\psi}_i$, and noise variance $\hat{\sigma}_a^2$ into Eq. (2.3) [1]. Note that the σ_a^2 term cancels. A point estimate, $\hat{\xi}$, is then found:

$$\hat{\xi} = \frac{\sum_{i=0}^{\infty} \hat{\psi}_i^2}{\sum_{i=0}^{D-1} \hat{\psi}_i^2} \tag{2.5}$$

Finally, an example of a non-quadratic performance measure is the closed-loop settling time, τ_s, defined in [19] as:

$$\psi_i^2 \leq \varepsilon^2, \; i \geq \tau_s \tag{2.6}$$

ε is a user-chosen parameter, below which ψ_i is considered to have settled. A point estimate for τ_s may be obtained by observing the estimated impulse response coefficients $\hat{\psi}_i$.

2.1.2 Normal approximations, linearization, and likelihood methods

This section describes existing work on estimating confidence intervals for the minimum variance performance indices. In [4] approximate moments are estimated for the performance index by assuming the disturbance variates (innovations) to be normally distributed. This approach is based on a linearization of the index, or Delta method, for calculating the variance. The variance of $\hat{\xi}$ for a sample size of N., can be shown to be approximated by [4]:

$$Var(\hat{\xi}) = (4/N)\xi^2 \left[\sum_{k=1}^{D-1} (\rho_k - \rho_{e,k})^2 + \sum_{k=D}^{\infty} \rho_k^2 \right] \tag{2.7}$$

where $\rho_{e,k}$, and ρ_k are the autocorrelations of the D step ahead prediction errors, and process output, respectively. The autocorrelations may be calculated directly from the time-series model parameters, see for example [2]. For practical use, the estimated model parameters may be substituted into Eq. (2.7). However, this method does not explicitly account for uncertainty in the model parameters. It has been shown that uncertainty in the parameters can lead to significant errors by using this method [2]. However, since the

estimated model parameters should converge to their true values as the sample size approaches infinity, the variance should also converge asymptotically [2]. A (1-α)100% confidence intervals may be estimated by assuming $\hat{\xi}$ is normally distributed:

$$\left(\hat{\xi} - z^{(1-\alpha/2)}\sqrt{Var(\hat{\xi})}, \hat{\xi} + z^{(1-\alpha/2)}\sqrt{Var(\hat{\xi})}\right) \tag{2.8}$$

where $z^{(1-\alpha/2)}$ is the 100(1-α/2)th percentile of a standard normal distribution. This interval is symmetric and does not account for the actual shape of the distribution, which can be skewed for small samples, or distorted when the index is close to its limiting value of one. This method is referred to as the normal method.

A method that does account for uncertainty in the model parameters was introduced in [3] where it was suggested that the variance of the performance index can be estimated by using the covariance matrix of the estimated time series parameters. This method is also based on a Delta approximation, and will be referred to as the linearization approach:

$$Var(\hat{\xi}) = f'(\theta)Cov(\theta)f'(\theta)^T \tag{2.9}$$

$$f'(\theta) = \left(\frac{\partial \xi}{\partial \theta_1}, \frac{\partial \xi}{\partial \theta_2}, \ldots \frac{\partial \xi}{\partial \theta_{np}}\right), \quad \theta = [\theta_1, \theta_2, \ldots \theta_{np}]$$

where θ is the vector of time series parameters. The covariance matrix, $Cov(\theta)$, is calculated via MATLAB's system identification toolbox. The most common methods for estimating $Cov(\theta)$ are based on linearization and asymptotic arguments [51]. The partial derivatives are numerically estimated by making small perturbations to the parameters [3]. Confidence intervals are estimated in the same fashion as Eq. (2.8). Methods based on Taylor expansions can become unreliable if the variables have large variance.

Generalized profiling has been used to estimate the confidence intervals for $\hat{\xi}$ [2]. In this method, no linearization is required, and the intervals may be non-symmetric. The intervals are estimated based on the asymptotic distribution of the likelihood ratio of the unconstrained and constrained likelihoods. To formulate the likelihood function it is first necessary to assume how the innovations are distributed. Assuming normally distributed innovations, the following procedure is used to estimate the (1-α)100% confidence interval (ξ_{low}, ξ_{high}) [2]:

1) Using some optimization routine, estimate the unconstrained set of model parameters θ, such that the likelihood is maximized, or for normally distributed variables such that

$$S(\theta) = \sum_{k=1}^{N} (y_t - \hat{y}_{t/t-1})^2 \tag{2.10}$$

is minimized. $\hat{y}_{t/t-1}$ is the one step ahead predictor. The set of parameters that minimizes Eq. (2.10) is $\hat{\theta}$. The index, $\hat{\xi}$, is then calculated based on $\hat{\theta}$.

2) Select a value of $d > \hat{\xi}$, and solve the constrained optimization problem to find the model parameters $\hat{\theta}^*$ that minimizes

$$S(\theta) = \sum_{k=1}^{N} (y_t - \hat{y}_{t/t-1})^2 \quad s.t. \quad \hat{\xi}^* = d \tag{2.11}$$

where $\hat{\xi}^*$, is then calculated based on $\hat{\theta}^*$.

3) Form the ratio:

$$N \ln\left(\frac{S(\hat{\theta}^*)}{S(\hat{\theta})}\right) \tag{2.12}$$

If $N \ln\left(\frac{S(\hat{\theta}^*)}{S(\hat{\theta})}\right) = \chi^2(1-\alpha/2)$, then $\xi_{high} = d$. $\chi^2(1-\alpha/2)$ is the $(1-\alpha/2)$ critical value of a chi-squared distribution with one degree of freedom. A similar procedure is used to find ξ_{low}, but constraints $d < \hat{\xi}$ are considered. Tutorial articles on generalized profiling may be found in [51, 52]. The procedure was programmed in MATLAB, using the function FMINCON to solve the constrained optimization. The prediction errors were calculated using the function PE, utilizing back casting (i.e. using the model to predict the initial conditions). One potential problem is that some less efficient estimators may only approximate the maximum likelihood estimates, which would result in the likelihood intervals being too narrow. Further, it assumes that the global minimum is found, but for nonlinear optimization there is no guarantee that this is true, i.e. the search algorithm may become stuck in local minima.

2.2 Residual based bootstrapping

Efron introduced bootstrapping in 1979 as a flexible method for estimating the sampling distributions of some statistic *T*. The statistic *T* is used to estimate the population parameter Φ. The original bootstrap method was based on a set of independent and identically distributed *(iid)* observations sampled from some distribution *F*. The sampling distribution of *T*(*F*) will be a function of the unknown population probability density. The basic idea of bootstrapping is to use the plug-in principle, which replaces the unknown population distribution with the empirical distribution $\hat{F}$ determined from the observations. The empirical distribution function places equal probability on each of the observations. Instead of repeating many experiments, which is equivalent to resampling from *F*, we resample from $\hat{F}$. Thus, in the "bootstrap world" $\hat{F}$ becomes the population distribution and we are concerned with

determining the sampling distribution of $T(\hat{F})$. In some simple cases, it is possible to analytically determine the sampling distribution $T(\hat{F})$. However, it is more common for the distribution of $T(\hat{F})$ to be approximated via simulation. This approach can be applied regardless of the complexity of the estimator, and essentially replaces difficult mathematical analysis with a computer intensive approach. Starting from a random sample size of N, a bootstrap sample of equivalent size is generated by sampling from the empirical distribution with replacement. The resulting bootstrap sample of size N is meant to mimic a random sample drawn from the original population. If B bootstrap samples are generated, the statistic of interest is then calculated for each of the B different bootstrap samples. This results in an empirical distribution for $T(\hat{F})$ allowing its moments and percentiles to be determined. Methods for relating the distribution obtained via bootstrapping $T(\hat{F})$ to the real world distribution $T(F)$ are described in the sequel. This thesis presents an empirical comparison of different bootstrap methods in estimating the bias, variance, and confidence intervals of the minimum variance benchmark. The closed-loop settling time is also studied. Good introductions to bootstrap techniques can be found in [53, 54]. An extensive bibliography of bootstrap applications is provided in [55].

This thesis utilizes residual bootstrap re-sampling methods to estimate the probability distribution of the minimum variance and settling time benchmarks. Bootstrapping methods were originally designed for *iid* data. However, the data collected in control situations will typical contain some serial correlation. To account for this correlation, the estimated residuals are resampled as opposed to resampling from the observed process outputs. The key idea behind the proposed bootstrap technique is explained through Figure 2-1 and Figure 2-2.

As shown in Figure 2-1, the closed-loop system is excited by a particular realization of the disturbance variates, a_t. This results in an observed set of output data, y_t. A time series model, $\hat{\psi}$, such as an ARMA is fitted to the output y_t. The output may have to be

differenced a number of times prior to the model fitting to ensure stationarity. The performance measure of interest, Φ, will be a function of the time series model parameters. A point estimate, $\hat{\Phi}$, of Φ is calculated from the time series model $\hat{\psi}$. The estimated residuals, $\hat{a}_1, \hat{a}_2, \hat{a}_3, \ldots$, are also calculated from the one step ahead prediction errors of $\hat{\psi}$. Once the time series model has been determined, the bootstrap simulated experiments may be performed as shown in Figure 2-2.

The residuals are re-centered to have a mean value of zero, and are then randomly sampled with replacement. Variables corresponding to a bootstrap sample are denoted by the traditional superscript *. A given bootstrap sample, $\hat{a}^{*i}$, is filtered through the impulse response model $\hat{\psi}$. This results in $\hat{y}_t^{*i}$, which is treated as a new set of observations. A new time series model, $\hat{\psi}^{*i}$, is then fit to $\hat{y}_t^{*i}$. Once the new time series model is found, an estimate for the control performance measure Φ may be calculated. The above procedure is repeated B times, for each of the bootstrap samples taken from the original estimated residuals. This results in B estimates of the performance measure, $\hat{\Phi}^*$, which can then be used to estimate the bias, variance and percentiles of $\hat{\Phi}$.

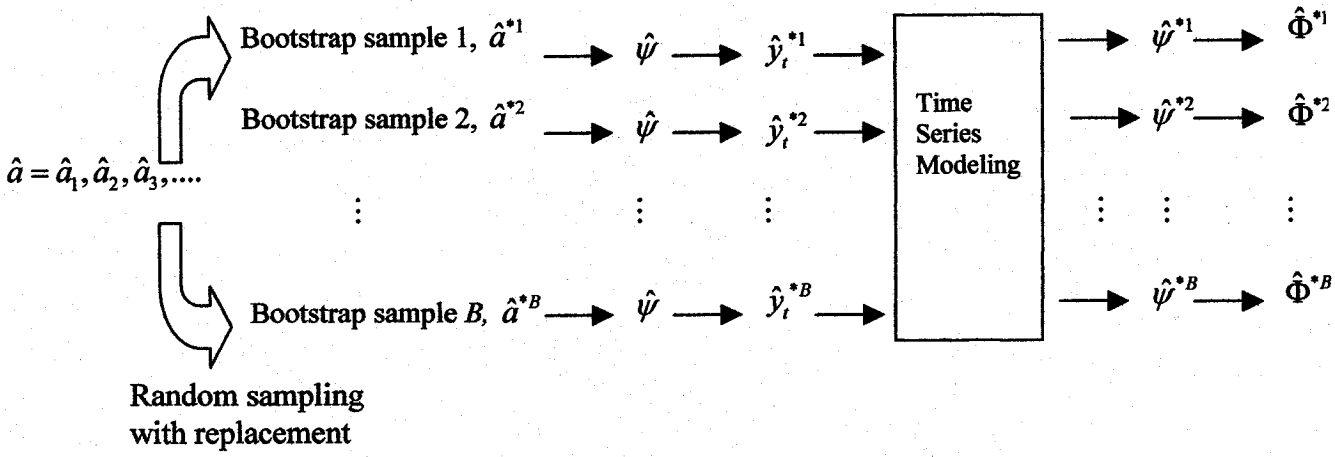

Figure 2-2. Bootstrap sampling, and estimation of the distribution for $\hat{\Phi}^*$.

There are two sources of error in the bootstrap method: The first is the simulation error resulting from the use of a Monte Carlo simulation to approximate $T(\hat{F})$. This error can be reduced by increasing B. However, this also increases the computational effort. Typically, values of B around 50-200 are suitable for estimating variance, and around 1000-2000 bootstrap samples are required for estimating confidence intervals [56]. Larger values of B are required for accurate confidence intervals so that good approximations of the distribution tails can be obtained. The second source of error is due to statistical error, which results from $\hat{F} \neq F$. One approach is to use a large sample size, in which case $\hat{F}$ will be a better approximation of F. Another method to handle this error is to use pivotal statistics. The distribution of pivotal statistics does not depend on any unknown quantities. Hence we would expect the distribution of a pivotal statistic to be the same under sampling from $\hat{F}$ or F.

2.2.1 Estimation of bias

The bootstrap estimates may be used to approximate the bias of the estimator $\hat{\Phi}$. The estimated bias can then be used for bias reduction. A simple bootstrap estimate for bias is:

$$bias^* \approx 1/B\sum_{j=1}^{B}\hat{\Phi}^{*j} - \hat{\Phi} \tag{2.13}$$

The estimated bias is simply the difference between the average of the bootstrap estimates, and the observed value. Hence, the bias corrected estimate, $\hat{\Phi}_{bc}$, is given by:

$$\hat{\Phi}_{bc} = 2\hat{\Phi} - 1/B\sum_{j=1}^{B}\hat{\Phi}^{*j} \tag{2.14}$$

A bias less than 0.25 standard errors can usually be ignored [56]. It should be remembered that bias* is only an estimate of the true bias. It is possible that the bias reduction will inflate the standard error of the estimator.

2.2.2 Estimation of variance

The variance of a statistic is a good overall measure of its uncertainty. *B* bootstrap samples from the estimated residuals are used to estimate *B* values of the statistic. The bootstrap estimate for the standard error, s.e, is:

$$se_B = \sqrt{1/(B-1)}\left(\sum_{i=1}^{B}[\hat{\Phi}^{*i} - 1/B\sum_{j=1}^{B}\hat{\Phi}^{*j}]^2\right)^{1/2} \tag{2.15}$$

2.2.3 Estimation of confidence intervals

There are several different methods for determining the confidence regions via bootstrapping. We will consider the standard, bootstrap-t, percentile, and the bias-corrected percentile (BC) methods.

Standard error method

If we assume the statistic *T* to be normally distributed, then we may use the bootstrap estimate of the standard error to find a confidence interval for Φ. Typically, analytical expressions for the standard error will not be available, and we will rely on the bootstrap approximation. The (1-α)*100% confidence interval is given by:

$$\left(\hat{\Phi} - z^{(1-\alpha/2)}se_B\right) \leq \Phi \leq \left(\hat{\Phi} + z^{(1-\alpha/2)}se_B\right) \tag{2.16}$$

where $z^{(1-\alpha/2)}$ is the 100(1-α/2)th percentile of a standard normal distribution. For the 95% confidence intervals, $\alpha = 0.05$, and $z^{(0.975)} = 1.96$. The advantage of this method is that the standard error can be calculated relatively quickly, and it is reasonable for statistics that are close to normal. The confidence interval may be adjusted for bias as follows:

$$\left(\hat{\Phi} - bias^* - z^{(1-\alpha/2)} se_B\right) \leq \Phi \leq \left(\hat{\Phi} - bias^* + z^{(1-\alpha/2)} se_B\right) \tag{2.17}$$

Typically, the statistic of interest is only approximately normally distributed for finite samples or may appear skewed. The normal method ignores the actual shape of the bootstrap distribution. Eqs. (2.16) and (2.17) should not be used for distributions that appear non-normal. The following methods generally improve the accuracy of the estimated intervals.

Bootstrap-t method

The bootstrap-t (also referred to as the percentile-t) method of generating confidence intervals is a generalization of the well-known student-t statistic. We could attempt to develop intervals for $\hat{\Phi}^* - \hat{\Phi}$, however experience has shown that by studentizing the statistic more stable confidence regions are produced [57]. This means that our bootstrap distribution of Φ^* will not be a strong function of the underlying distribution from which the bootstrap samples are drawn (*i.e.*, it is an approximately pivotal statistic). This is important since the estimated residual distribution is only an approximation of the true residual distribution. Let,

$$T^* = (\hat{\Phi}^*(b) - \hat{\Phi}) / se_B(b)^* \tag{2.18}$$

Then the 100(1-α)% confidence interval is approximated as:

$$\hat{\Phi} - T^{*(1-\alpha/2)} se_B \leq \Phi \leq \hat{\Phi} - T^{*(\alpha/2)} se_B \tag{2.19}$$

Note that a nested bootstrap will be necessary for estimating the standard deviation, $se_B(b)^*$, for each of the bootstrap samples. This is performed by bootstrap sampling from each of the original bootstrap samples. As reported in [56], around 25 nested bootstrap samples are sufficient for estimating the standard error. However, around 1000 bootstrap samples are required for accurate estimation of confidence intervals. This requires 25,000 cycles for generating bootstrap-t confidence intervals. Another disadvantage is that the bootstrap-t intervals can behave erratically, and are sensitive to the scale used. Experience has shown that for the bootstrap-t method to work well the variance should be stabilized [58]. If this is not the case, then the variance stabilizing bootstrap-t method [56, 58] could be used, but won't be considered here. An advantage of the bootstrap-t method is that it takes into account the actual shape of the bootstrap distribution.

Percentile and bias corrected percentile methods

In the percentile method, the bootstrap percentiles are used to determine the limits of the confidence interval. This method is predicated on the existence of a normalizing monotone transformation for $\hat{\phi} = g(\hat{\Phi})$, such that it has a mean $g(\Phi)$ and a variance that does not change with $g(\Phi)$. It is interesting to note that this transformation need not be known in order to use the percentile method. The percentiles are obtained by ordering all of the bootstrap estimates for Φ. Consider B bootstrap estimates $\hat{\Phi}^{*1}, \hat{\Phi}^{*2}, \hat{\Phi}^{*3}, ..., \hat{\Phi}^{*B}$, then the α-percentile is given by $P_\alpha = (B+1)\alpha^{\text{th}}$ ordered value. Thus the (1-α)*100% confidence interval is:

$$(P_{\alpha/2}, P_{1-\alpha/2}) \tag{2.20}$$

This method requires the existence of a transform, which converts the bootstrap distribution to a symmetric distribution. However, frequently there is no such transformation. Hence, this method works best when the distribution of $\hat{\Phi}^*$ is approximately symmetric [59]. In addition, this method assumes that the estimator is unbiased. To account for biased estimators, the bias corrected (BC) method is used. The BC method assumes a monotone transformation exists such that $\hat{\phi}^* - \hat{\phi}$ and $\hat{\phi} - \phi$ are $N(-b\sigma, \sigma^2)$. Let Γ be the cumulative function of a standard normal distribution, then the BC confidence interval is:

$$(P_{\alpha_1}, P_{\alpha_2}) \tag{2.21}$$

where, $\alpha_1 = \Gamma(2z_0 + z^{\alpha/2})$, $\alpha_2 = \Gamma(2z_0 + z^{1-\alpha/2})$, and $z_0 = \Gamma^{-1}(\frac{\#(\hat{\Phi}^*(b) < \hat{\Phi}}{B})$.

The symbol # represents the number of bootstrap estimates which are less than $\hat{\Phi}$.

In all of the following examples, the 'true' distributions of $\hat{\xi}$ and $\hat{\tau}_s$ are determined via a Monte-Carlo simulation with 50,000 repetitions. Note, the bootstrap method only utilizes the information contained in a single set of observations of sample size *N*. The Monte-Carlo results provide a basis of comparison for evaluating the performance of the bootstrap procedures. Time series identification was performed using the ARMAX function in MATLAB.

2.2.4 Examples

In all of the following simulated examples, the 'true' distributions of $\hat{\xi}$ and $\hat{\tau}_s$ are determined via a Monte-Carlo simulation with 50,000 repetitions. Note, the bootstrap

method only utilizes the information contained in a single set of observations of sample size *N*. The Monte-Carlo results provide the 'gold-standard', to which other results will be compared.

Consider the following case, which is taken from [29]:

$$G(z^{-1}) = \frac{0.33}{1-0.67z^{-1}},\ D = 4 \tag{2.22}$$

$$Gd = \frac{1-0.4z^{-1}}{1-0.67z^{-1}} \tag{2.23}$$

$$C(z^{-1}) = \frac{0.7-0.47z^{-1}}{0.33-0.1z^{-1}-0.23z^{-4}} \tag{2.24}$$

$C(z^{-1})$ is the transfer function of the controller. The theoretical value of ξ is 1.52. Unless otherwise stated the innovations are distributed as $N(0, 0.36)$. A (5,0,5) ARIMA model is used to model y_t. Figure 2-3 provides a comparison of the sampling distributions of $\hat{\xi}^*$ determined via bootstrapping to the true sampling distribution of $\hat{\xi}$ for sample sizes ranging from 50 to 1000. The distribution of $\hat{\xi}^*$ appears to closely mimic the true distribution of $\hat{\xi}$. It is important to note that the bootstrap distribution may be centered at a different location than the true distribution, providing an estimate for the bias via Eq. (2.13).

The bootstrap estimates of bias stabilized after 100-200 bootstrap replications (Figure 2-4). The bias estimates behaved quite erratically. This result is consistent with [56], which warns that the variance of the bias estimate can inflate the variance of the bias corrected estimator. In Table 2-1, the column of averaged bias estimates were determined by calculating the bias estimate ($B = 200$) for 20 different realizations and then averaging the results. On average the bias estimates are good for intermediate

sample sizes (N <500), but it does not appear to be able to resolve the smaller biases which occur for sample sizes smaller than 500. Thus the bootstrapping method may not provide a proper correction to the performance index.

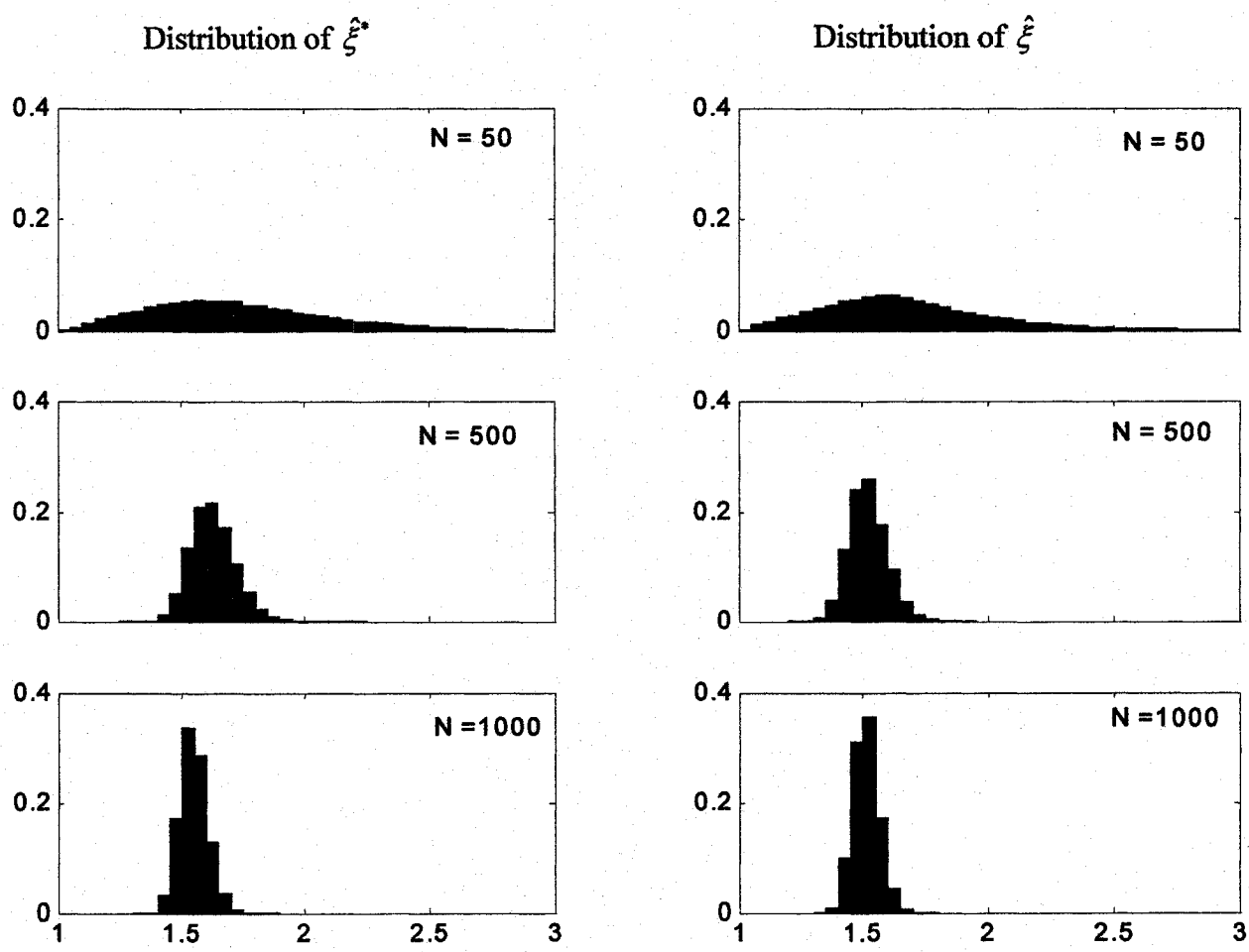

Figure 2-3. Comparison of the probability distributions of $\hat{\xi}^*$ and $\hat{\xi}$ for different sample sizes.

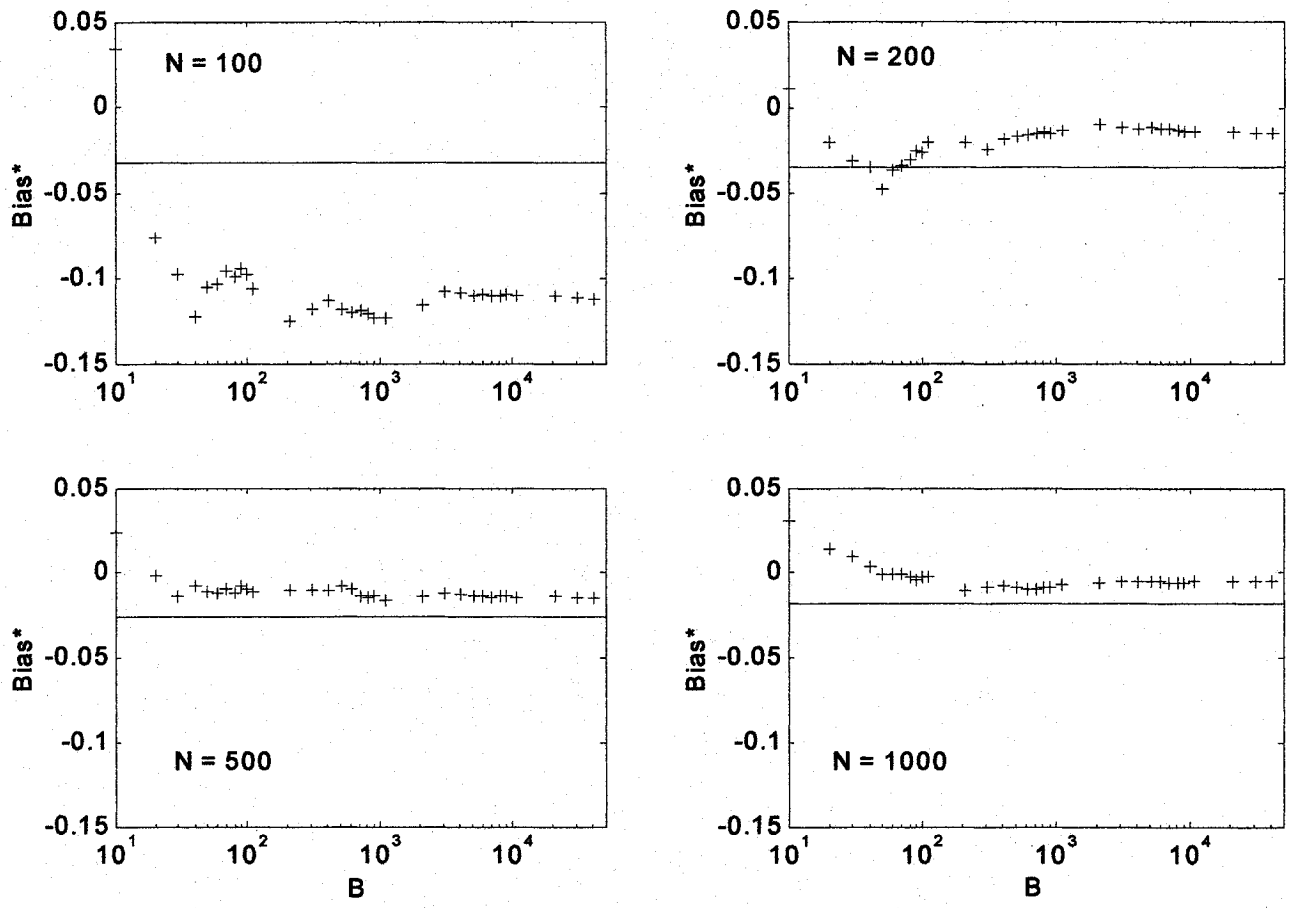

Figure 2-4. Bootstrap estimate of bias.

Table 2-1: Bootstrap estimates of bias.

N	True Bias	Bootstrap	Average Bootstrap
50	0.22	0.28	0.40
100	0.11	0.03	0.13
200	0.041	0.065	0.045
500	0.0041	0.018	0.022
1000	0.0036	0.010	0.0085

The bootstrap provided more reliable estimates for the variance, and 95% confidence interval. In the following comparisons, all methods utilized the same realization for each N. The bootstrap took around 100-200 replications for the estimated variance to stabilize (Figure 2-5). All of the methods provided reasonable estimates of the variance for larger sample sizes (Table 2-2). However, for $N = 50$ both the normal and linearization methods underestimated the true variance by an order of magnitude. The parametric bootstrap was performed by sampling from a mean zero normal distribution with a variance equal to the estimated residual variance. There was little difference between the parametric and nonparametric bootstrap estimates. Note that it is important to obtain good estimates from small samples since in practice the process data may only be stationary over short intervals.

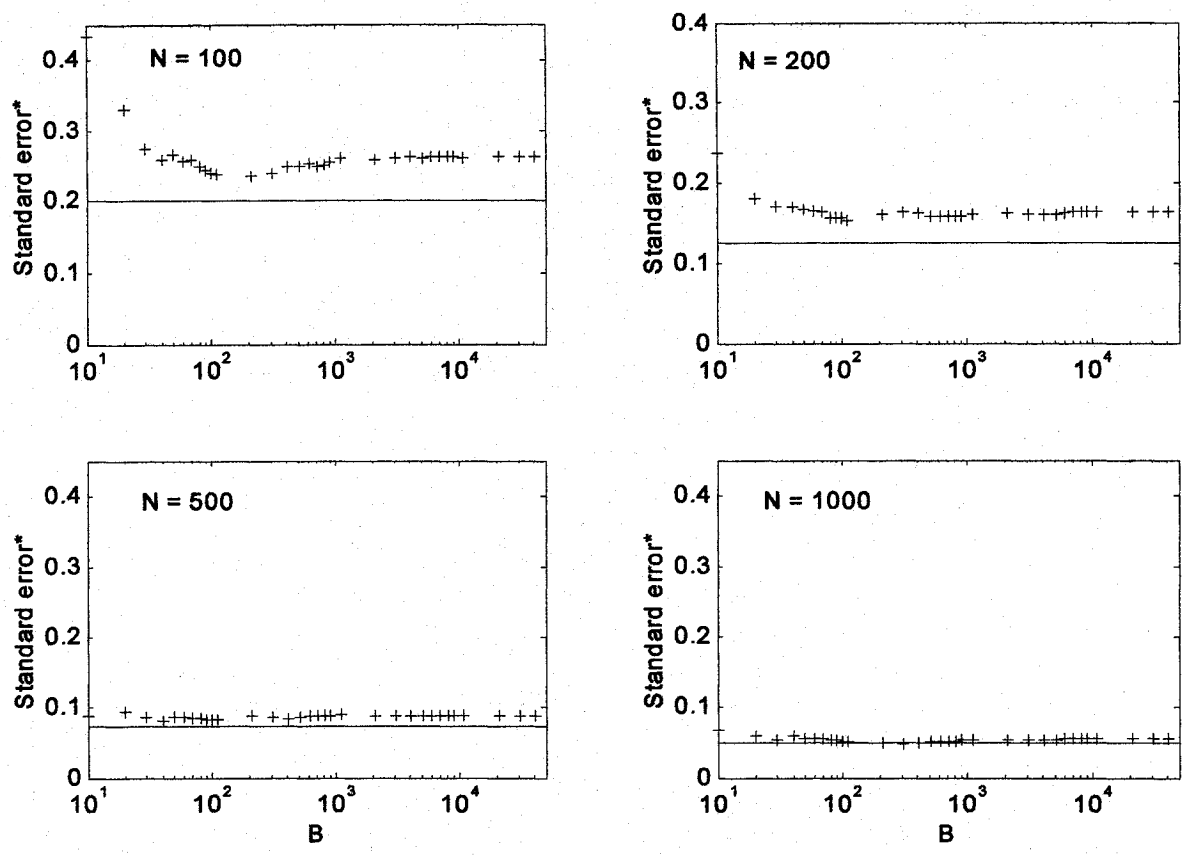

Figure 2-5. Bootstrap estimate of standard error.

Table 2-2. Comparison of variance estimates.

N	True Variance	Bootstrap Parametric	Bootstrap Nonparametric	Normal	Linearization
50	0.42	0.56	0.55	0.038	0.046
100	0.052	0.091	0.087	0.034	0.052
200	0.017	0.029	0.029	0.019	0.020
500	0.0058	0.0082	0.0080	0.0067	0.0071
1000	0.0027	0.0032	0.0031	0.0028	0.0029

Next, the different bootstrap methods of estimating a 95% confidence interval for $\hat{\xi}$ are compared. A sample size of $N = 1000$ was used, and for the bootstrap-t method 100 nested bootstrap replications were used to estimate the standard error. Figure 2-6 shows the upper and lower limits of the bootstrap 95% confidence intervals. Since these bootstrap confidence intervals only approximate the true interval, the actual coverage will not be exactly 95%. The term 'coverage' refers to the actual percent of times an interval (generated in a similar fashion) will contain the true value of ξ under repeated sampling. The actual coverage of the bootstrap intervals is plotted in Figure 2-7. All of the bootstrap methods improve as B is increased, and around 2000-4000 bootstrap replications are required to get stable estimates of the confidence intervals. The bootstrap-t and bias corrected percentile methods are generally considered more accurate than the other bootstrap intervals considered in this paper. The true coverage after 4000 bootstrap replications was 96.1% for both the bootstrap-t and bias corrected percentile methods, respectively. In this case, it is hard to justify the extra computational burden imposed by the bootstrap-t method. It is important for the intervals to have accurate coverage if they are to be used in hypothesis testing to detect significant changes in performance.

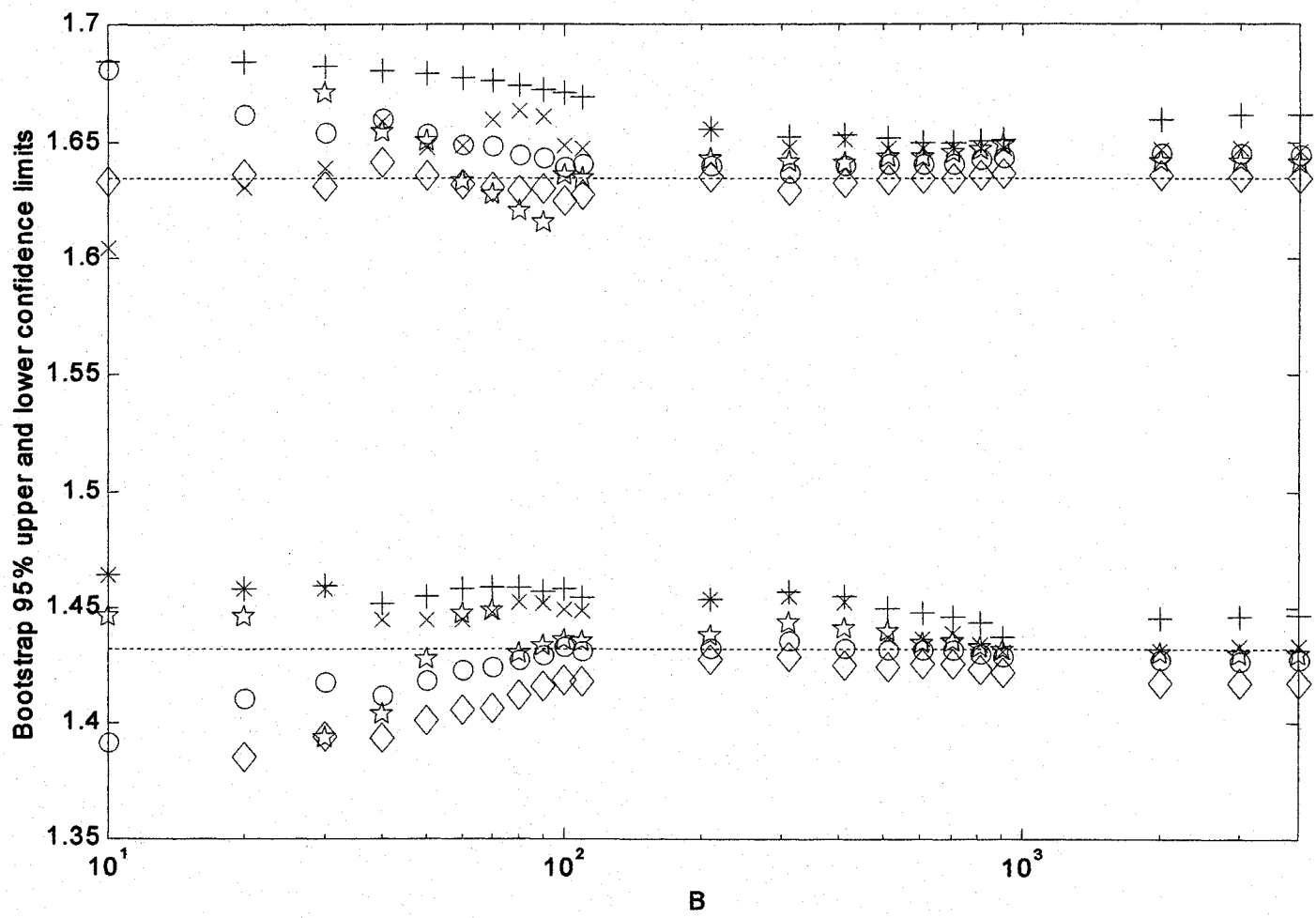

Figure 2-6. Bootstrap estimates of the upper and lower limits of the 95% confidence interval for ξ. The dotted lines represent the true upper and lower limits. (+: percentile method, x: bias corrected percentile, ○: normal method, ◊: bias corrected normal method, ☆: bootstrap-t method.)

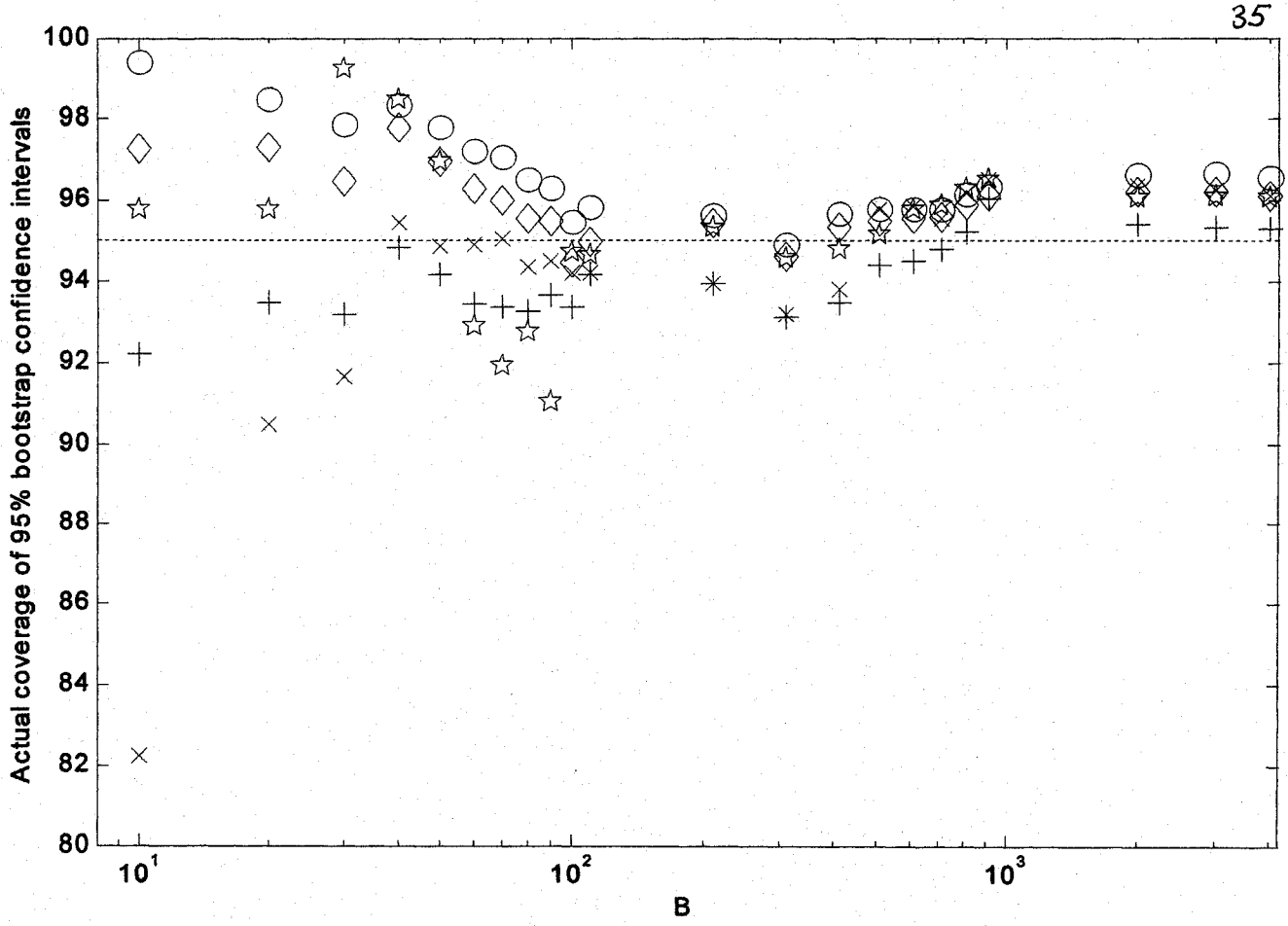

Figure 2-7. Actual coverage of the 95% bootstrap confidence intervals for ξ. (+: percentile method, x: bias corrected percentile, ○: normal method, ◊: bias corrected normal method, ☆: bootstrap-t method)

The deviation from normality is apparent in the quantile-quantile (Q-Q) plots of $\hat{\xi}^*$ (Figure 2-8). The percentile and bootstrap-t methods can accommodate deviations from normality. The Q-Q plots suggest that our estimator is asymptotically normal, but deviates from normality for smaller sample sizes. Methods based on normal intervals should be used with caution for smaller samples.

The bootstrap (bias-corrected percentile, and bootstrap-t) confidence intervals are compared to the normal, linearization, and likelihood methods (Table 2-3). For larger sample sizes all methods provides similar limits that closely match the true upper and lower bounds. For $N = 50$ both the normal and linearization methods appear to provide accurate limits, but these should be treated with suspicion since they are based on an inaccurate variance estimate. If the normal and linearization methods had estimated the variances exactly, they would result in intervals (1, 2.78) and (1.17, 2.07) for $N = 50$ and 100, respectively. Comparing these results to the true intervals, it is clear that the assumption that $\hat{\xi}$ is normally distributed is not appropriate. Even for $N = 50$, the bootstrap method provides a reasonable estimate of the limits, while the likelihood interval has shifted further to the right of the true limits. The bootstrap-t method is slightly more accurate than the bias-corrected percentile method, at the cost of increased computation. Parametric versions of bootstrap-t and bias corrected percentiles confidence intervals were similar to the non-parametric counterparts.

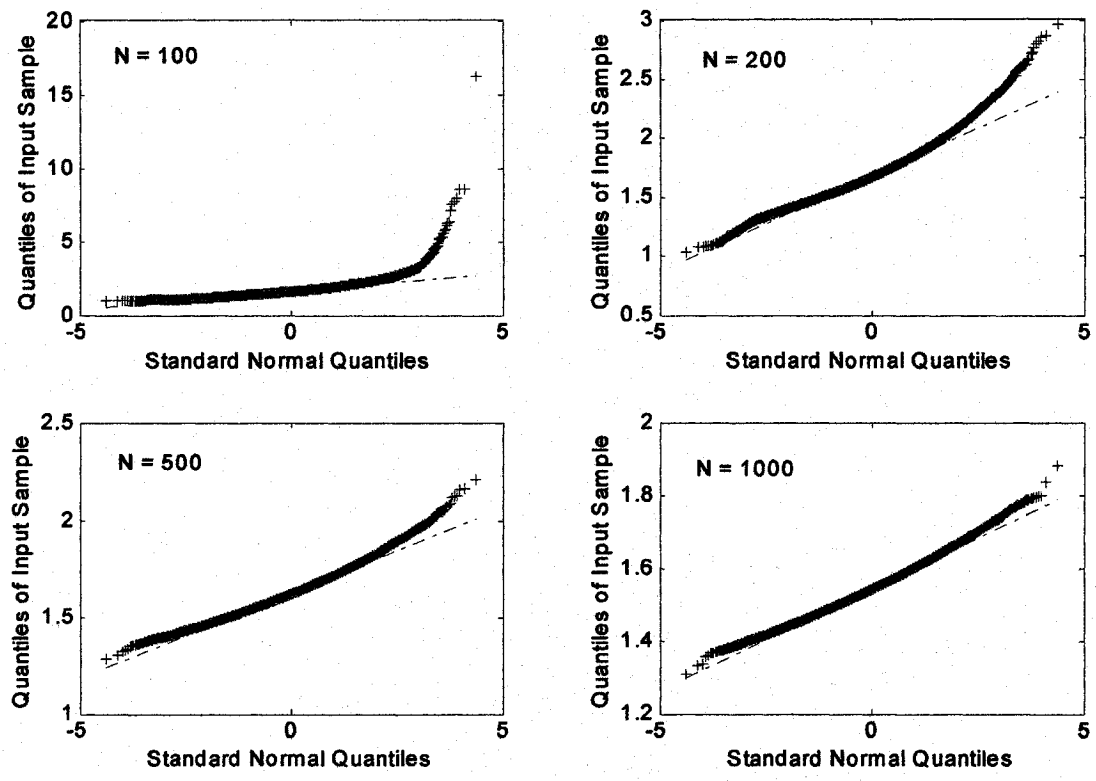

Figure 2-8. Quantile-quantile plots with $B = 50{,}000$ to test the normality of $\hat{\xi}^*$.

Table 2-3. Estimated 95% confidence intervals.

N	$\hat{\xi}$	True 95% Interval	Bootstrap-t	Bias corrected percentile	Normal	Linearization	Likelihood
50	1.51	(1.0, 1.90)	(1.0, 1.97)	(1.05, 2.09)	(1.13, 1.89)	(1.09, 1.93)	(1.21, 2.21)
100	1.62	(1.0, 1.90)	(1.22, 2.21)	(1.19, 2.39)	(1.26, 1.99)	(1.18, 2.07)	(1.21,2.31)
200	1.62	(1.30, 1.81)	(1.32, 1.89)	(1.32, 1.92)	(1.36, 1.89)	(1.34, 1.90)	(1.37, 2.00)
500	1.61	(1.44, 1.74)	(1.44, 1.77)	(1.46, 1.78)	(1.45, 1.77)	(1.44, 1.77)	(1.43,1.73)
1000	1.54	(1.43, 1.63)	(1.43, 1.64)	(1.43, 1.65)	(1.43, 1.64)	(1.43, 1.64)	(1.42,1.66)

Lower limits less than 1.0 are truncated to 1.0.

Consistent with [2], it was found that the methods were robust for slight deviations of the innovations from normality. There was little change in the performance when the innovations were distributed as a student-t distribution with three degrees of freedom. However, when the innovations were distributed uniformly on the interval [-0.5, 0.5] the sample size at which the normal and linearization methods failed to reasonably estimate the variance increased to $N = 100$. In this case, the true variance was 0.066, and the bootstrap, normal, and linearization estimates were 0.061, 0.017, and 0.024, respectively. The true 95% confidence interval was (1.0, 1.60), and the bias-corrected percentile bootstrap and likelihood intervals were (1.02, 1.62) and (1.11, 1.85), respectively. The bootstrap method provided a better estimate of the interval.

Confidence intervals may be calculated by all of the methods. Increased confidence in the results can be obtained if all methods provide essentially equivalent results. For small sample sizes if the methods provide different results, it is suggested that the bootstrap result will be more reliable. Certainly, if the residuals deviate strongly from

normality (a testable hypothesis), then the non-parametric bootstrap should be used. The advantage of the linearization approach is that the calculations can be performed relatively quickly. However, it should only be used when the index is distributed normally. A few hundred bootstrap replications could be performed to check if this is a reasonable assumption. When using the linearization approach it is important to keep in mind the limitations of the method used to estimate the covariance matrix of the model parameters [51]. Just as most statistical methods perform poorly with small samples, eventually so will the bootstrap. However, the bootstrap has performed well even with the relatively small sample size of $N = 50$. The likelihood technique should be limited of course to maximum likelihood estimators, or close approximations thereof. This highlights one further advantage of the bootstrap method; it can be applied to any computable estimator, maximum likelihood or otherwise.

Using the same system as described by Eq. (2.22) to (2.24), the bootstrap variance of the closed-loop settling time was examined. The closed-loop settling time, τ_s, will be defined by selecting $\varepsilon = 0.3$. Thus $\tau_s = 7$. Any settling times greater than 300 were rejected in both the Monte Carlo and bootstrap simulations. The nonparametric bootstrap estimates of the variance required anywhere from 2000-50,000 replications for result to stabilize (Table 2-4), which is significantly higher than the minimum variance index. The bootstrap provided reasonable estimates of the variance. It may be of interest to interpolate the settling time between sampling intervals, thereby allowing the settling time estimator to become differentiable (numerically) with respect to the model parameters. In this case, a Delta method approximation for the variance could be examined.

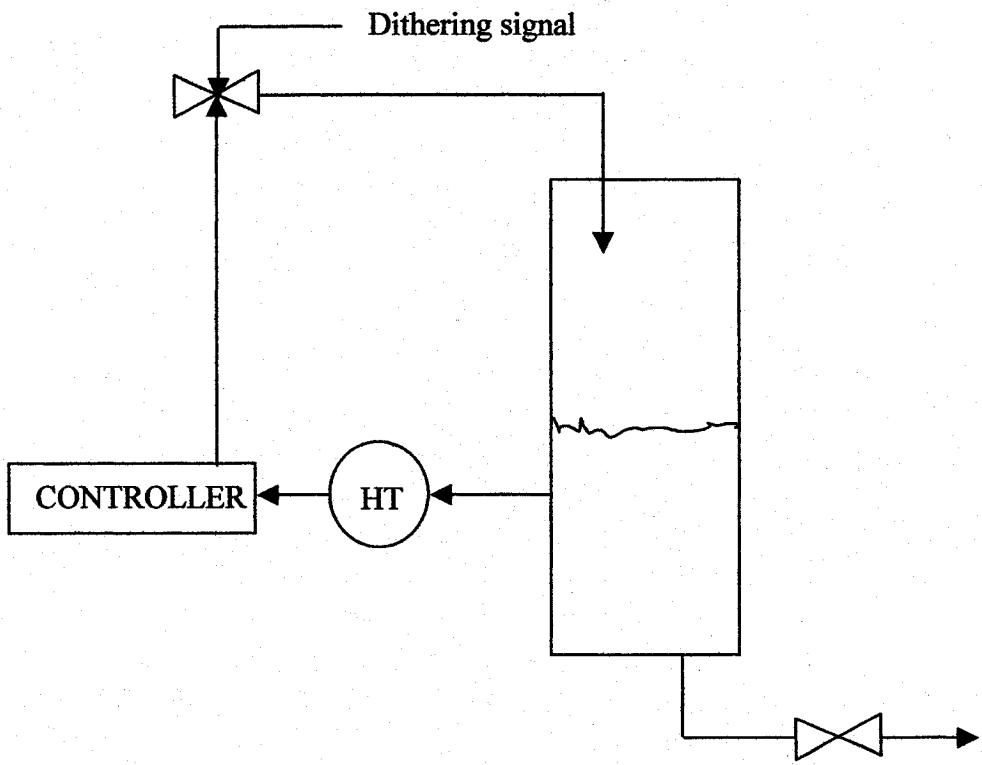

Figure 2-9. Experimental setup of liquid level control system.

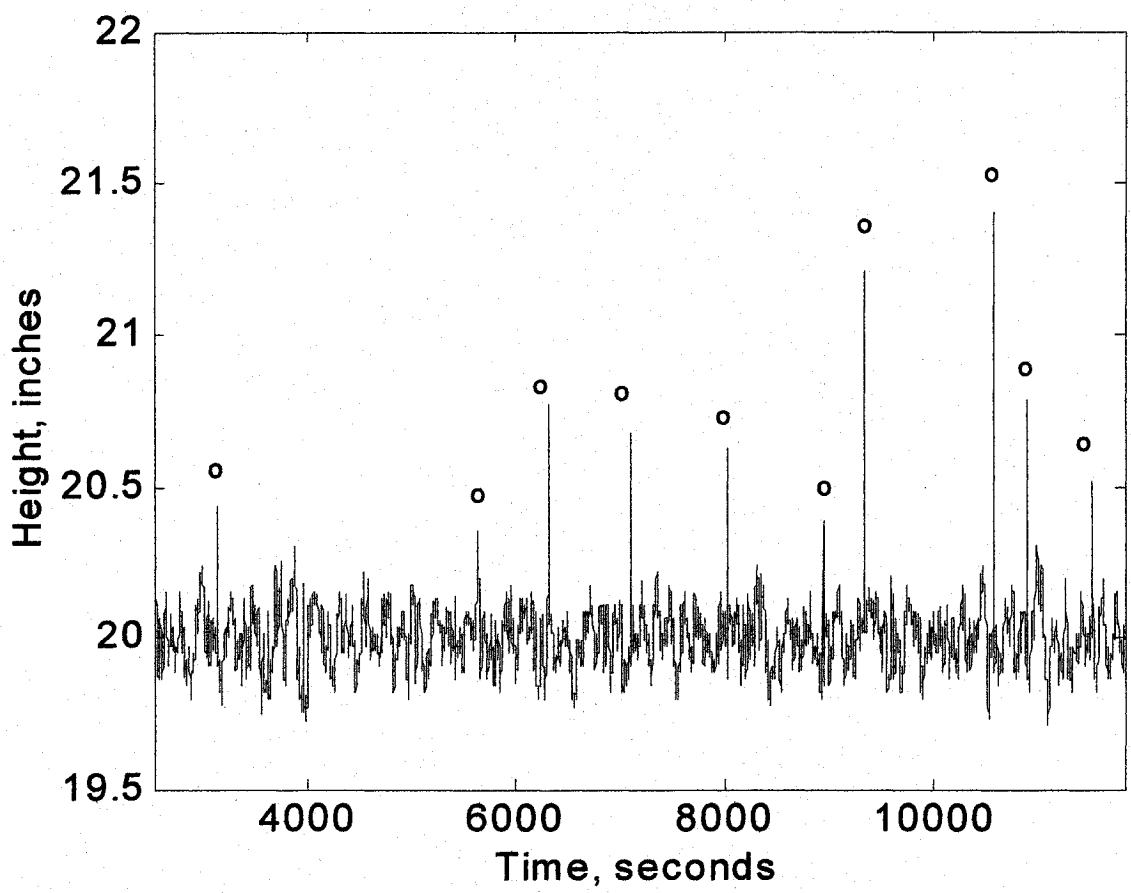

Figure 2-10. Experimental data from liquid level controller.

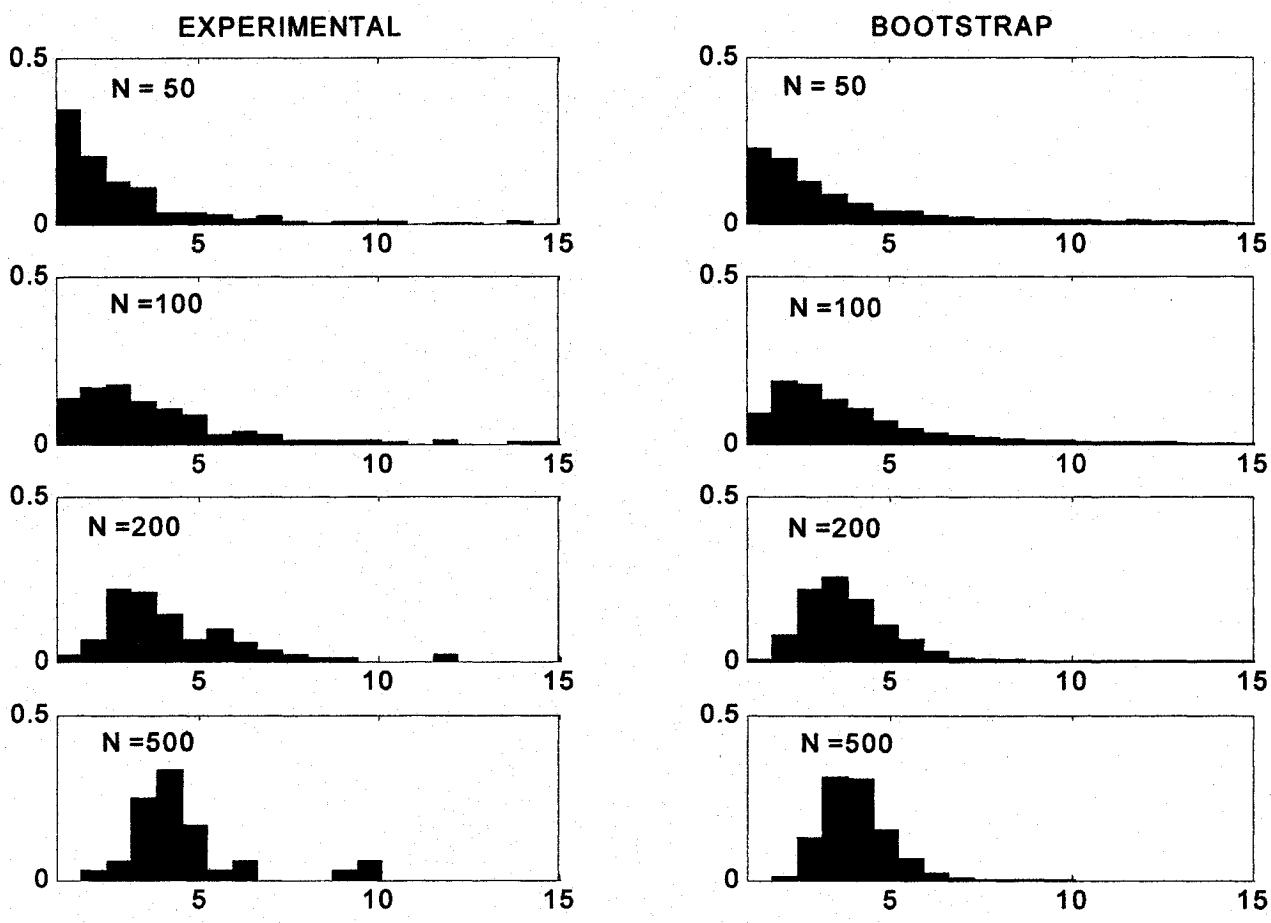

Figure 2-11. Bootstrap and experimental distribution of minimum variance index for liquid level controller.

2.3 Handling uncertainty in the model order

The sampling distribution of an estimator represents the variation of results under hypothetical repeated experiments. Since the process is random, by definition each experiment under identical conditions will lead to a different performance index. In earlier studies, the same model order has been used for each imaginary repeated experiment. This assumption is not realistic, since in practice the user must use some criteria (e.g. AIC) to select the model order. Therefore, the model order will behave randomly. In this thesis it is shown, through Monte Carlo simulation, that this extra uncertainty translates into a significant effect on the uncertainty of the minimum variance performance index. In addition, it is shown that the bootstrapping framework, easily allows for consideration of uncertainty in the model order.

By using the same example as given by Eq. (2.22) to (2.24), a sample of size N = 200 was generated. For simplicity, the AIC was used to select the optimal AR model. It was found that the AIC indicated a model order of ARIMA (9,0,0). A Monte Carlo simulation was used to determine the sampling distribution under the assumption that for each imaginary experiment, the user always selects a model structure of ARIMA (9,0,0). The Monte Carlo simulation was then repeated allowing for the selection of the optimal AR model with orders from one to twenty. For both of the Monte Carlo simulations 10,000 replications were produced. The resulting normalized distributions of the minimum variance performance index are shown in Figure 2-12. The variance of the index under the ARIMA (9,0,0) was 0.013, with a mean value of 1.35. However, when the model order uncertainty is considered, the variance almost doubled to 0.023, and the mean changed to 1.40. Therefore the uncertainty in the model order significantly increased the variance, and slightly reduced the bias of the performance index. The histogram of model orders is shown in Figure 2-13.

The earlier methods for estimating the uncertainty of the performance index have not considered the uncertainty of the model order, and it is not clear how this additional uncertainty can be incorporated into these methods. In contrast, the bootstrap framework allows the inclusion of the uncertainty in model order (Figure 2-14). In the bootstrap algorithm, after each new realization $\hat{y}_t^{*1}$ has been generated, the AIC is determined for model orders ranging from one to twenty, and the optimal structure is selected. Using 4000 bootstrap replications it was found that the bootstrap estimate of the variance was 0.031. For a sample index of 1.4, the true 95% confidence interval is (1.16, 1.75), while the bias corrected percentile method provided a reasonable estimate of (1.14, 1.67). The other bootstrap estimates of the 95% confidence limits are shown in Figure 2-15. Due to the extra computational burden of model order selection, the bootstrap-t was not considered.

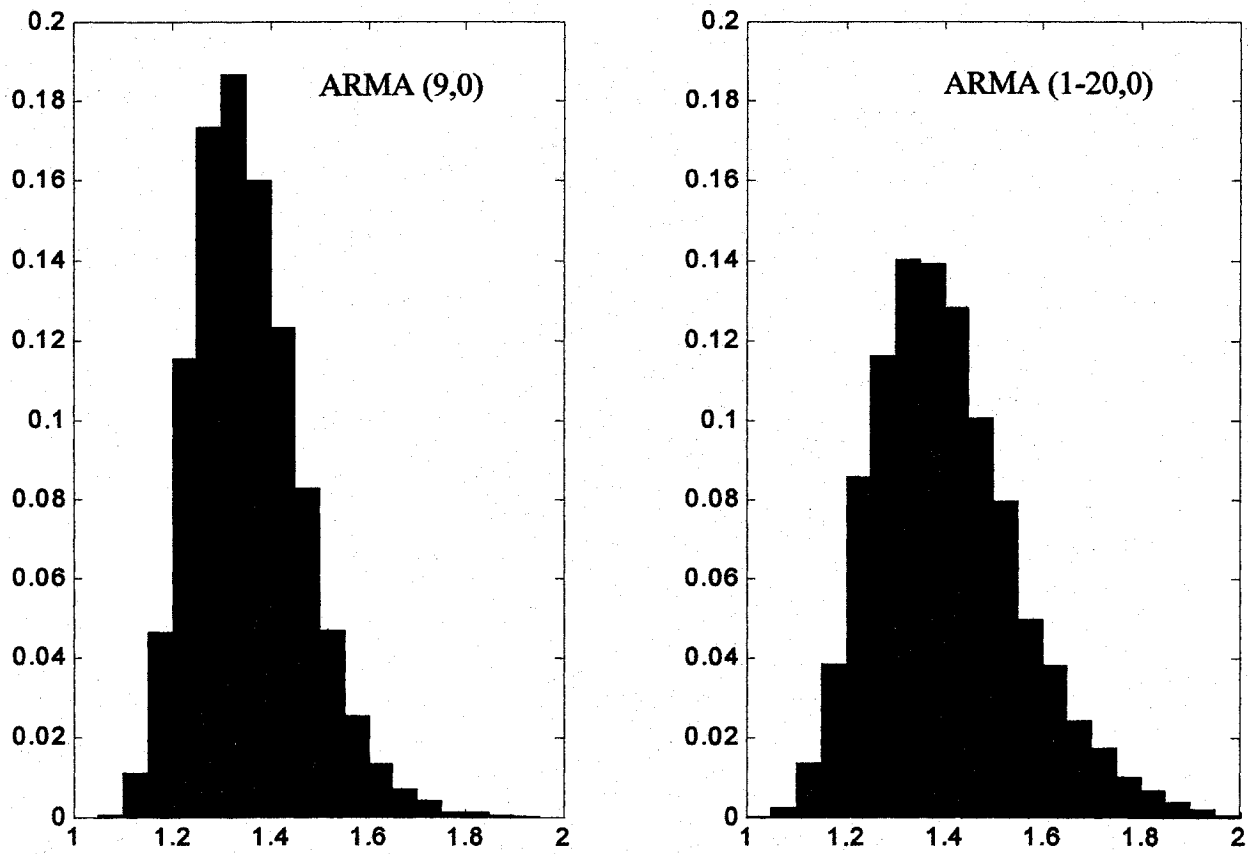

Figure 2-12. Distribution of minimum variance index for ARMA (9,0) and flexible model order ranging from 1-20.

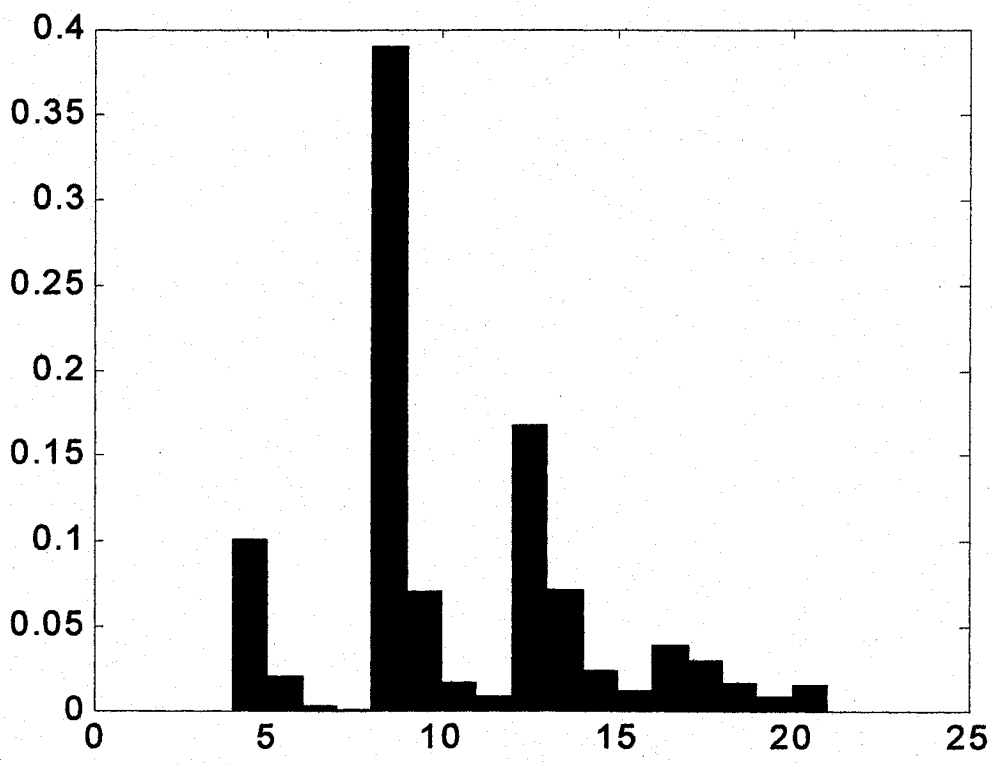

Figure 2-13. Histogram of AR model orders.

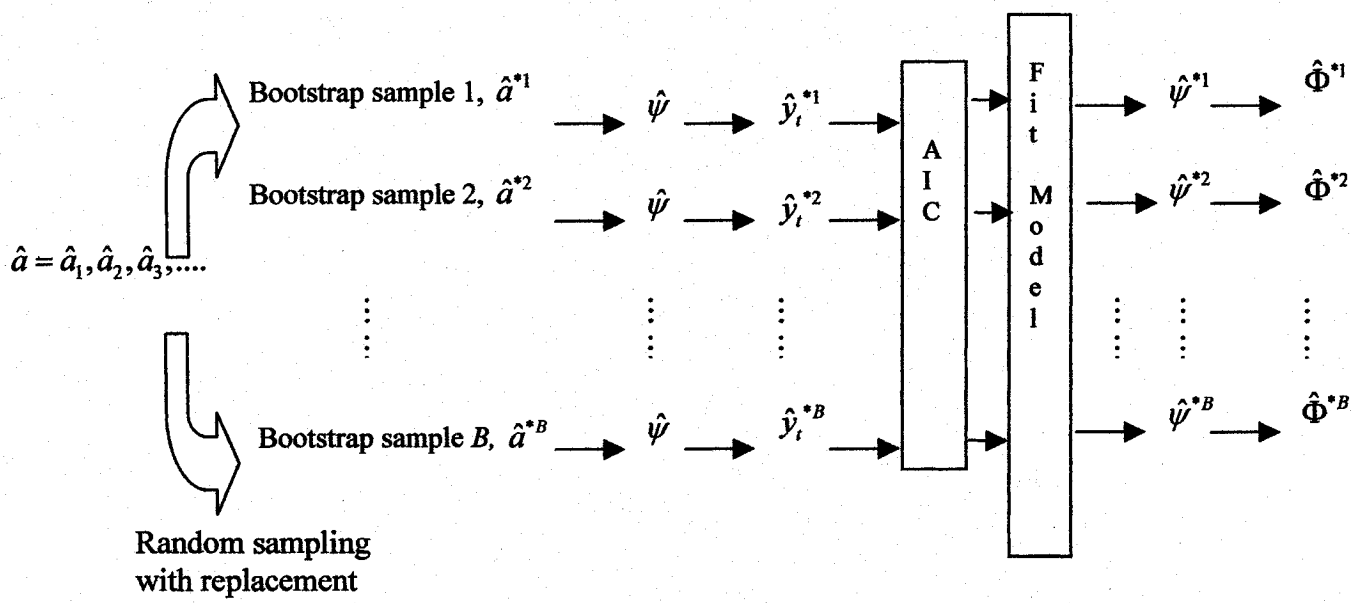

Figure 2-14. Bootstrap sampling, and estimation of the distribution for $\hat{\theta}^*$.

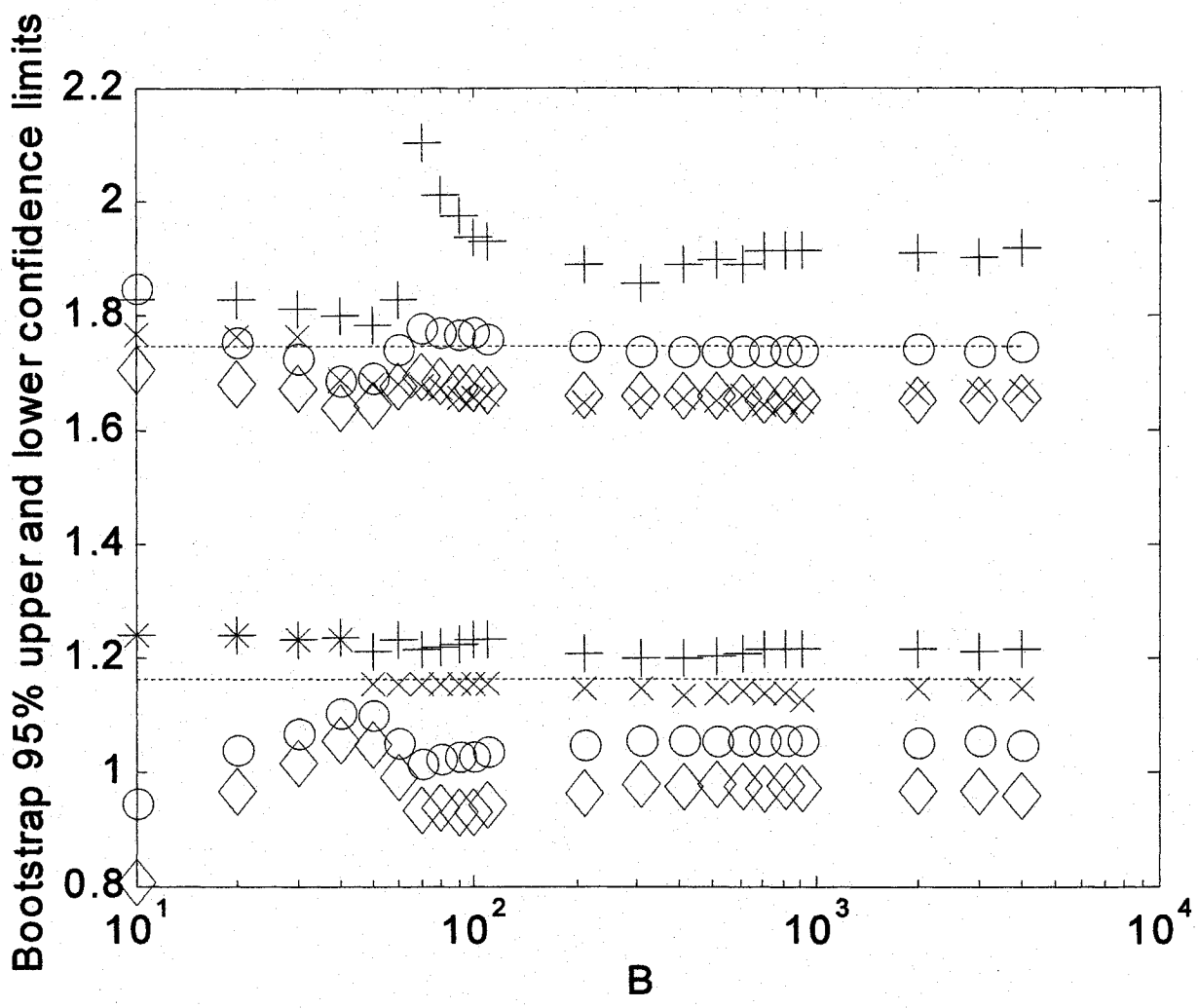

Figure 2-15. Bootstrap estimates of the upper and lower limits of the 95% confidence interval for ξ. The dotted lines represent the true upper and lower limits. (+: percentile method, x: bias corrected percentile, ○: normal method, ◊: bias corrected normal method)

CHAPTER 3 MODEL PLANT MISMATCH (MPM)

It is well known that the explicit use of a process model in designing a controller can significantly improve controller performance. Naturally, the performance of model-based controllers (MBC) will be intimately linked to the quality of the process models. If some performance measure indicates poor performance, e.g. [11, 33, 34, 36, 48, 60], the root cause must be diagnosed. Some common causes of poor performance are: model plant mismatch (MPM), the physical limitations imposed by the process interactor matrix and spectral characteristics of the disturbances [33, 60], poor tuning, instrumentation/equipment failures. MPM may arise from a poorly identified model, and/or changes in process or operating conditions. It is important to detect MPM since it can lead to degradation of performance and instability. Further, more accurate models allow the controllers to be tuned more aggressively, resulting in better performance. Therefore, this chapter will focus on diagnosing the presence of MPM

It has been noted that while there is significant work in assessing the performance of controllers, there has been less research in the area of diagnosing the root causes of poor performance [11]. Many of the methods used to diagnosis MPM provide a 'yes/no' answer with respect to MPM, and do not specify the parameters or the subset of models that are mismatched in the case of multivariable systems. Univariate systems were considered in [6],where it was shown that if the disturbance is white noise then cross correlation between the prediction error and input u will be zero if the model is perfect. However, this is not true when the more realistic case of colored noise (i.e. white noise passed through a linear filter) is considered. Therefore, set-point dithering has been used to test for correlation between the set-point and prediction error [6]. A significant correlation would indicate the presence of MPM. Their method is applicable to SISO systems. In this thesis, I have extended their method to evaluate MPM for multivariable

systems. A multivariable chi-squared test on the output and prediction errors has been suggested to help diagnosis MPM in [7]. In [8, 9] the multivariable complementary sensitivity functions are identified in the frequency domain, and are compared to their design specifications. Any significant differences would imply the presence of MPM. A prediction index, which compares the minimum achievable prediction error variance to the actual variance, has been proposed in [10]. A benchmark relevant for model predictive controllers proposed in [11] compares the achieved to designed cost functions. A significant change in this benchmark would be due to changes in the process or unmeasured disturbance characteristics. Hence, it does not distinguish between process mismatch and disturbance model mismatch. Since it is common practice to use a simple step disturbance model in the controller design, this method would always indicate mismatch even if the process models were perfect (unless the disturbance was actually a step disturbance). This thesis considers methods that are insensitive to disturbance model mismatch. In [12] the control loop is opened so that a disturbance model may be identified. The loop is re-closed, and a time series model is fit between the white noise driving the disturbance and the output error. If the time series model for the output error is equal to the disturbance model then no MPM is present. Any difference between the two models would result from MPM or changing disturbance characteristics. This method has the advantage that it doesn't require any external excitation, but it is sensitive to changing disturbances, and the control loop must be initially opened to identify the disturbance model. They briefly mention that different subsets of the inputs can he held constant to determine which specific inputs are mismatched. The idea of holding certain subsets of the inputs to aid in identifying which models are mismatched was also proposed by Webber and Gupta [61] and is further explored in this thesis. By adding a perturbation signal to the set point, the cumulative sum change detectors were used to determine if performance degradation was due to deterministic disturbances or control relevant systems changes [62]. An interesting series of papers that detect abrupt changes in model parameters in the presence of varying disturbance dynamics has been presented

[63, 64] . A two-model divergence algorithm is used in [64]. In [63] output error models were used to reduce the sensitivity of the detection scheme to time varying disturbances.

Several different approaches are adopted in this thesis. Of course careful parametric modeling [14, 65] can be applied to the closed loop, and the resulting model can be compared to the existing model that would provide an explicit description of the MPM. This type of identification requires expertise and is time consuming. The main idea behind all the methods is to avoid a rigorous parametric identification, and to use non-parametric techniques that can help indicate if such a careful parametric modeling is required. In some cases methods are presented for correcting specific parameter mismatches.

In the next sections, an overview of the two most common methods of controller design that explicitly utilize a process model, model predictive control (MPC) and internal model control (IMC) is presented.

3.1 Model predictive control

MPC was introduced by industry in the 1970's to handle constrained multivariable control problems. Previously the state-space methods, such as Linear Quadratic or Linear Gaussian Regulation (LQR or LQG), of the 1960's were very popular in the aerospace industry. However these methods were not readily adopted by the processing industry [66, 67]. In the processing industry, the optimal operating point is frequently at the constraint boundaries. Previous control methods could not safely account for the multivariable interactions and constraints so the operating points had to be a safe distance away from the optimal point. To maximize profit in an uncertain and dynamic market it is important to have a control method which safely allows the operating point to be moved close to new optimal operating points while satisfying operating constraints. Therefore MPC's ability to handle multivariable interactions and constraints was very attractive to industry. MPC technology has been successfully applied in many industrial settings,

mainly in refining or chemical processing plants. As of 1999 it was estimated that MPC was used in over 4500 different applications[68]. Several industrial applications of MPC are described in [69]. The power of MPC is evident when applied to multivariable systems. The largest reported application of MPC as of 2003 is to a 603×283 system [68]. It is claimed that part of MPC's popularity is due to the fact that it is easy to understand [69]. Good overviews of MPC are found in [67, 68, 70-76]

MPC algorithms make explicit use of a model to predict the future response of the process over a finite horizon. In this thesis, I only consider MPC algorithms that utilize linear models for prediction. Most MPC algorithms use a receding or moving horizon strategy: An optimal input trajectory is determined on line over a finite control horizon at each control instant. Only the first move of the input trajectory is actually applied to the process, and the whole procedure is repeated at the next control instant. Figure 3-1 graphically depicts the receding horizon method. The free response represents the effect of past moves. The free response is adjusted by adding a bias term, which usually is the difference between the measured value of y (y_k) and the predicted value of y ($\hat{y}_k$) at the current control interval. This bias term is sometimes referred to as the adjustment term. This procedure of updating the prediction assumes any mismatch is the result of a step disturbance, and provides feedback for the controller. The current and future control moves from k to $k+M-1$, where M is the control horizon, are determined by minimizing the sum of squared errors between the set point and the predicted response of y over the prediction horizon N_p. Since the open-loop optimization is solved online at each control instant a certain amount of time is required for numerical processing. The chemical processing industry is dominated by slow processes, which allows adequate time to determine the solution. However, the sampling time should be small to address disturbances. To help reduce the on-line computational effort, simplified model predictive control (SMPC) has been proposed [77]. In SMPC, a control horizon of one (M=1) is considered and the error is usually minimized only at a single point over the prediction horizon.

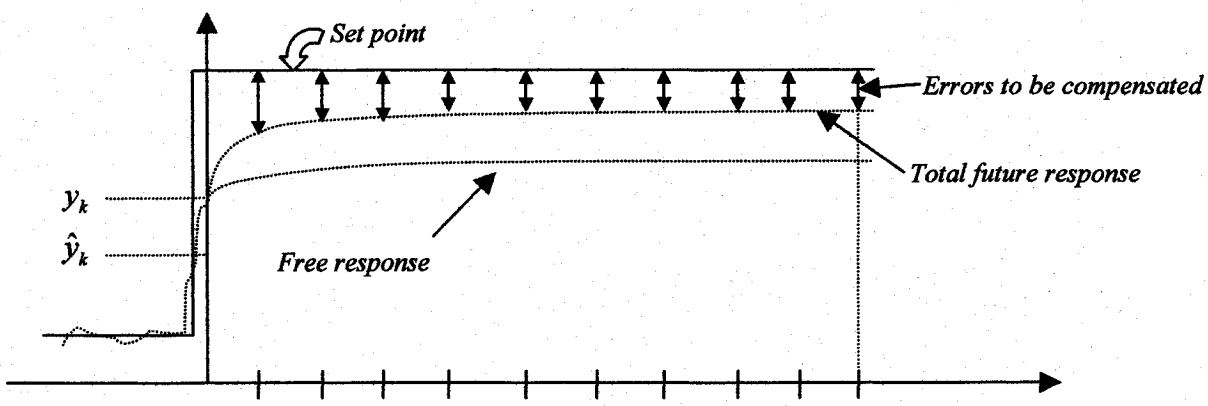

Figure 3-1. Receding horizon with prediction horizon of N_p

The most common forms of MPC used in industry are based on Dynamic Matrix Control (DMC) [78] , and Identification and Command (IDCOM) also referred to as Model Algorithmic Control (MAC) [66]. DMC uses step response models to predict the future behavior of the process, and IDCOM utilizes impulse response models. While the state space approach is popular in academic research, not many commercial packages utilize the state space approach [68].Both methods assume that a step-disturbance is responsible for any mismatch between the predicted and observed output. While this is a simplistic assumption, it does lead to offset free tracking for step changes in the set point or disturbance. In practice the constraints have usually been handled through ad-hoc methods [68] .

To explicitly handle constraints, the DMC problem was reformulated as a quadratic programming (QP) problem referred to as QDMC [79, 80]. QP problems are attractive since they are convex, and can be solved easily with existing methods such as interior point or active set methods. IDCOM was also reformulated as a QP problem called IDCOM-M [81]. DMC has also been formulated as a linear programming (LP) problem called LDMC [82]. Further details may be found in [83, 84]. A detailed overview of the various commercial MPC packages is provided in [68].

3.2 Dynamic matrix control

This thesis considers the unconstrained form of DMC. First the unconstrained SISO DMC is described, followed by the MIMO case. Further details on the derivation of the DMC control law may be found in [78, 79]. The truncated step-response model used for prediction is:

$$y_{k+l} = \sum_{i=1}^{Ns-1} H_i \Delta u_{k+l-i} + H_N u_{k+l-N} \tag{3.1}$$

where:

k is the present control instant

H_i is the i^{th} step response coefficient,

y_{k+l} is the predicted value of y at instant $k+l$

u_k is the input at instant k,

and N_s is the number of time steps for the process to settle.

Let M be the control horizon, then the $N_p \times M$ dynamic matrix A is defined as:

$$A = \begin{bmatrix} H_1 & 0 & & 0 \\ H_2 & H_1 & & \vdots \\ H_3 & H_2 & & H_1 \\ \vdots & \vdots & & \vdots \\ H_N & H_{N-1} & \cdots & H_{N_p-M+1} \end{bmatrix} \tag{3.2}$$

The control action that minimizes the predicted error in the least squares sense is:

$$\Delta u = (A^T A)^{-1} A^T e \tag{3.3}$$

where e is a N_p dimensional vector of the predicted errors between the set point and the free response over the prediction horizon. Δu is a M dimensional vector containing the

present (Δu^1) and future control moves (Δu^j, $j = 2,3,...M$). Only the first control move, Δu^1, is implemented, and the rest are discarded. This is an unconstrained solution. Move suppression is introduced to reduce the magnitude of changes in the manipulated variables and improve robustness to modeling errors. Move suppression is implemented by increasing the diagonal elements of the A^TA matrix.

The MIMO solution is developed naturally from the SISO case. Let N_y and N_u be the number of outputs and inputs respectively. The MIMO dynamic matrix is:

$$A = \begin{bmatrix} A_{11} & A_{12} & & A_{1N_u} \\ A_{21} & A_{22} & & \vdots \\ A_{31} & A_{32} & & \vdots \\ \vdots & \vdots & & \vdots \\ A_{N_p1} & A_{N_p2} & \cdots & A_{N_pN_u} \end{bmatrix} \tag{3.4}$$

where A_{ij} is the SISO dynamic matrix between the i^{th} output and j^{th} input. The MIMO DMC control law is:

$$\Delta u = (A^T A)^{-1} A^T e \tag{3.5}$$

However, now e and Δu are stacked column vectors of dimension $N\,xN_y$ and $M\,x\,N_u$ respectively. More specifically:

$$\Delta u = [\Delta u_1\ \Delta u_2 \cdots \Delta u_{N_u}]^T \tag{3.6}$$

$$e = [e_1\ e_2 \cdots e_{N_y}]^T \tag{3.7}$$

where $\Delta u_i = (\Delta u_i^1, \Delta u_i^2, ..., \Delta u_i^M)$, with $i = 1,2,...N_u$, and $e_i = (e_i^1, e_i^2, ..., e_i^N)$, with $i = 1,2,...N_Y$.

3.3 Internal model control

IMC provides a general framework for MPC. The relationship between IMC and DMC, MAC, and other control methods is described in [85]. They also provide an analysis of the stability for IMC structures. Figure 3-2 shows the IMC structure in which the model $\hat{G}$ and the plant G are in parallel. The signal ε is fed back to the controller, and is comprised of two parts: the model mismatch $\theta = G - \hat{G}$, and the disturbance v:

$$\varepsilon = \theta + v \tag{3.8}$$

The IMC structure is particularly useful when studying model mismatch. IMC controllers are designed by inverting the minimum phase portion of the process, applying a filter for robustness and by ensuring that the resulting controller is proper [85-87]. Some important properties of IMC controllers include: zero offset for step loads/setpoint changes if the controller gain is the inverse of the process model gain, and for stable processes with perfect models the closed loop is stable if the controller is stable.

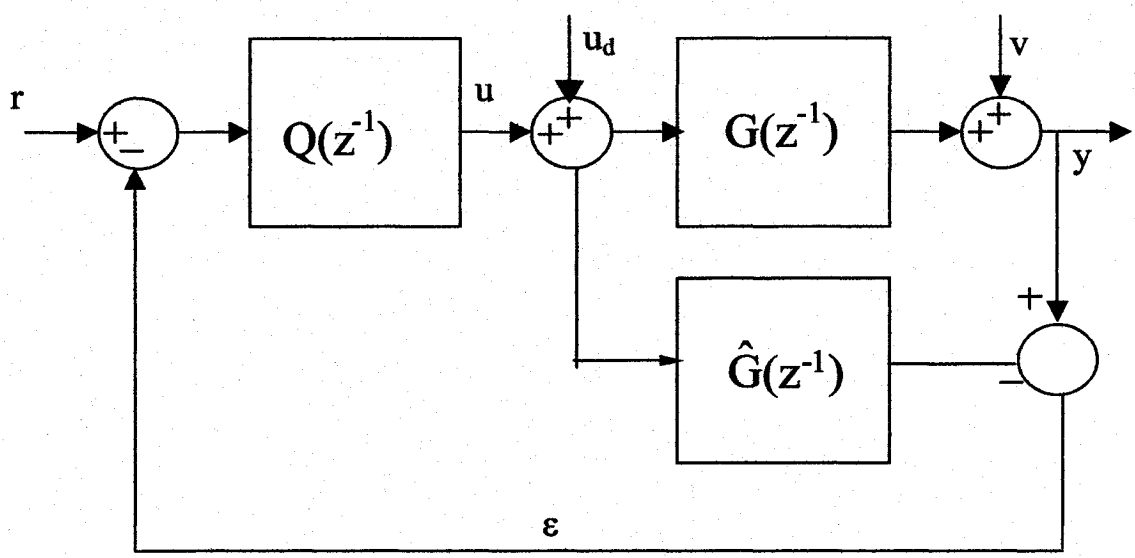

Figure 3-2. IMC structure.

Let θ be the model mismatch matrix, defined by the relation $\theta = G - \hat{G}$. The closed loop transfer functions are:

$$\varepsilon = (I + \theta Q)^{-1} \theta Q r + (I + \theta Q)^{-1} \theta u_d + (I + \theta Q)^{-1} v \tag{3.9}$$

$$u = (I + Q\theta)^{-1} Q r - (I + Q\theta)^{-1} Q\theta u_d - (I + Q\theta)^{-1} Q v \tag{3.10}$$

$$y = G(I + Q\theta)^{-1} Q r + G(I - (I + Q\theta)^{-1} Q\theta) u_d + (I - G(I + Q\theta)^{-1} Q) v \tag{3.11}$$

where r is the setpoint, u_d is the input dither signal, v is disturbance, y is output, ε is the prediction error, Q is the controller, and finally G and $\hat{G}$ are the process and model transfer function matrices respectively. Equations (3.9) to (3.11) are valid for univariate or multivariate systems with the proper replacement of scalar elements with vectors or matrices.

3.4 The difficulty in detecting model-plant mismatch

From Figure 3-2 one may assume that the presence of model mismatch may be detected by simply monitoring the prediction error signal, ε. If there is no disturbance then the signal ε does represent the effects of model mismatch. For systems with no noise, examination of the signal ε would directly indicate the presence of model mismatch. However, disturbances are always present, thus there are a couple of difficulties in this approach:

- A large ε does not indicate a large θ since it may be the result of a large disturbance signal, or it may be due to a small model error with a large magnitude u.

- The traditional open-loop validation method of testing for correlation between the input u, and prediction error ε will fail because the feedback structure causes the input to be correlated to ε even when the model is perfect.

- For MIMO systems, there is a model between each of the inputs and the outputs. So there are $N_u \times N_y$ models between the different input-output pairs and any one of these may be mismatched.

In a MIMO system ε is a vector, $\varepsilon = [\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_{N_y}]^T$ whose elements represent the error signal for each output. These elements represent the cumulative error in a particular output due to all of the inputs. Under conditions of no disturbance (or low amplitude disturbances) ε would provide direct information about which output is mismatched. For example if $\varepsilon_1 = \varepsilon_2 = 0$ and $\varepsilon_3 \neq 0$ then we would know that models for y_1 and y_2 contain no mismatch, but at least one of the models for y_3 is mismatched. This does not provide enough information to determine which specific model used to describe y_3 is a problem. We wish to determine which specific input-output pairs of y_3 are

mismatched. Since linear models are used the superposition property applies for $\forall i = 1,2,...N_y$:

$$y_i = \hat{G}_{i1}u_1 + \hat{G}_{i2}u_2 + \hat{G}_{i3}u_3 + ... + \hat{G}_{iN_u}u_{N_u} \tag{3.12}$$

where $\hat{G}_{ij}$ is the model between the i^{th} output and the j^{th} input. In DMC $\hat{G}_{ij}$ is parameterized as a step-response model.

The following sections presents the various methods developed in this thesis for diagnosing MPM. The first section considers univariate systems, and the second section considers multivariate systems.

3.5 Univariate systems

This section examines the transient response of the prediction error to set point changes for SISO systems. This work was motivated by our previous work [13], where we noticed that certain patterns can be associated with parameter mismatch in first order plus dead time process models. The purpose is to check if the certain patterns in the prediction error may be used to diagnosis the model mismatch, without resorting to more complicated closed-loop identification methods. It is found that gain mismatch leads to a distinctive pattern, however it may be difficult to distinguish between dead-time and time constant mismatch. Also, this section shows that the patterns can depend on the controller tuning. For simplicity and clarity, no disturbances are considered at this time. However, the presence of disturbances will require that the set point change is large enough to provide a good signal to noise ratio in the prediction error, which is a standard requirement for any transient analysis. This can be addressed by using a small magnitude dithering signal on the set point and using a correlation analysis. First order plus dead

time (FOPDT) models will be considered since a large number of processes in the chemical and petroleum processing industry may be well approximated using such a model.

Consider the following control system:

$$G = \frac{e^{-4s}}{10s+1} \qquad \hat{G} = \frac{\hat{K}e^{-\hat{D}s}}{\hat{\tau}s+1} \qquad Q = \frac{\hat{\tau}s+1}{\hat{K}(\lambda s+1)} \tag{3.13}$$

where $\hat{K}$ is the model gain, $\hat{D}$ is the model delay, and $\hat{\tau}$ is the model time constant. Let the triplet $m = (\hat{K}, \hat{\tau}/10, \hat{D}/4)$ define the mismatch. The IMC tuning parameter is selected as $\lambda = 4$. Figure 3-3, Figure 3-4, and Figure 3-5 show the response of the prediction error to pure gain, time constant and dead time mismatch respectively.

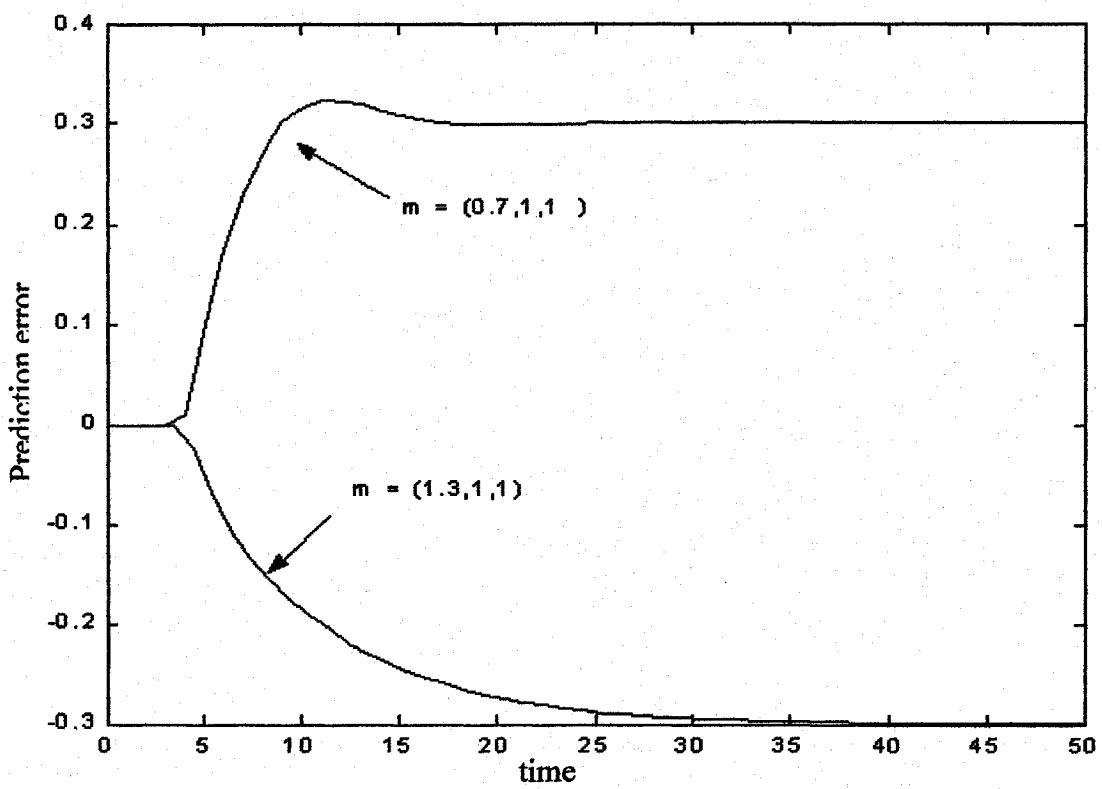

Figure 3-3. Prediction error pattern for ± 30% gain mismatch.

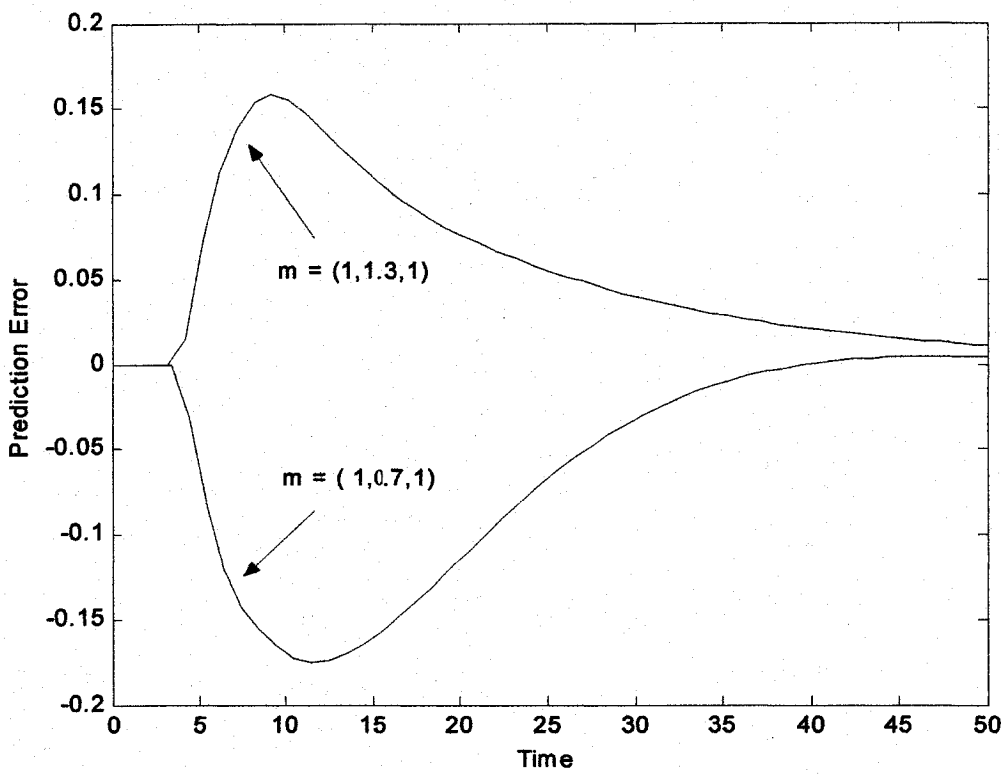

Figure 3-4. Prediction error pattern for ± 30% time constant mismatch.

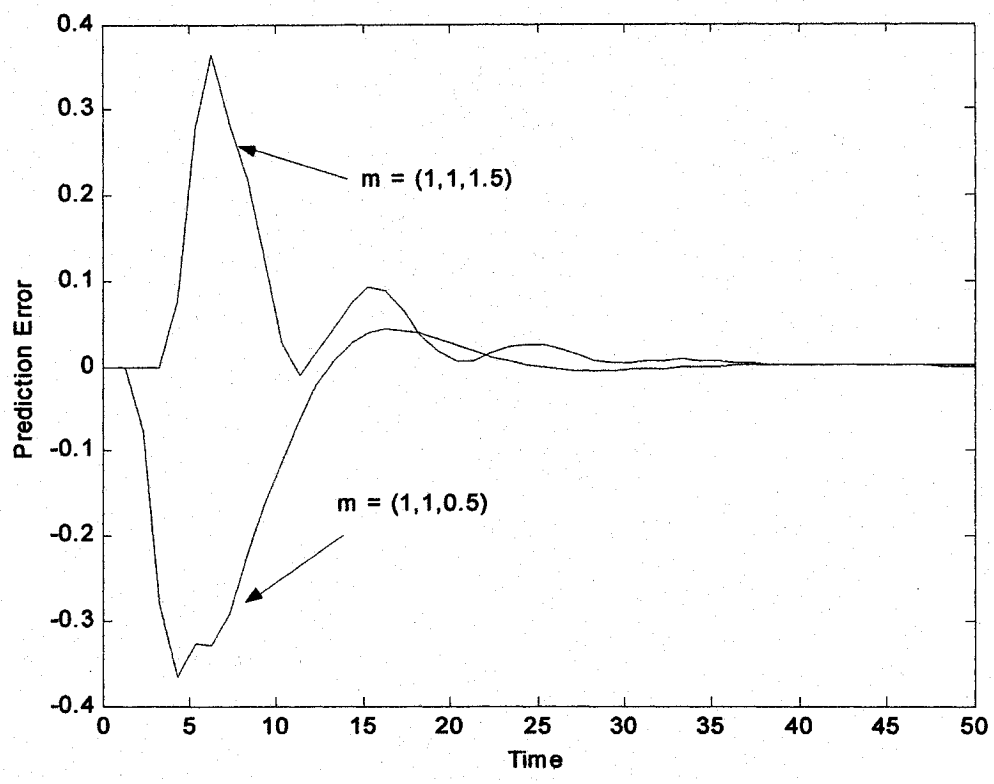

Figure 3-5. Prediction error pattern for ± 50% dead time mismatch.

It is clear that if only gain, time constant or dead time mismatch is present, then the transient analysis of the prediction error provides information if the respective parameter is over or underestimated. And this provides the user with a simple indicator if the parameter should be increased or decreased. For example, if the time constant is over estimated then the prediction error shows a slow increasing trend, which then decays to zero. If the time constant was underestimated then it would show an initial negative trend. However, the exact patterns also depend on how conservatively the controller is tuned. If the controller is very sluggish ($\lambda = 15$) then the pattern for dead time mismatch can appear similar to that for time constant mismatch (Figure 3-6). Hence it may be difficult to distinguish between time constant and dead time mismatch unless the initial patterns are generated for the controller tuning being used. However, as shown below, gain mismatch has a special characteristic that may be detected even in the presence of other mismatched parameters. Also, a simple method for correcting the dead time mismatch is presented.

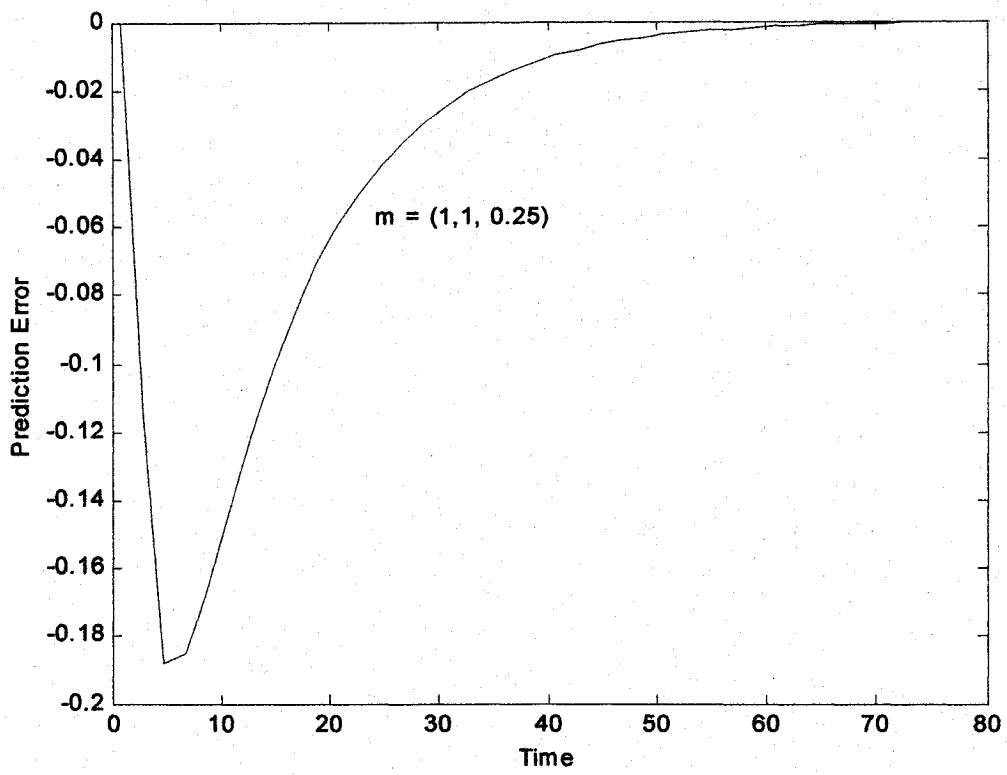

Figure 3-6. Prediction error pattern for dead time mismatch with detuned controller.

Consider the SISO internal model control (IMC) structure in Figure 3-2. All transfer functions are assumed to be proper, and linear time invariant (LTI). From Equation (3.9) it is clear that if there is no MPM, then the transfer functions ε / r and ε / u_d will be zero. Identification of these transfer functions will obviously provide information concerning the MPM. Under closed-loop operating conditions the identification of these two transfer functions is effectively open-loop. This is because the input r and u_d can be chosen to be statistically independent of the disturbance v. This allows the use of traditional open-loop identification methods such transient response analysis or correlation analysis to detect MPM under closed-loop conditions. Since the controller Q and process model $\hat{G}$ are known, the process G may be back calculated from ε / r or ε / u_d. It may be noted that $(G - \hat{G})$ is simply the additive uncertainty term used in robust control. Once the transfer function ε / r is identified, then Equation (3.14) may be used to update the model.

$$\left(\hat{G}\right)_{NEW} = \frac{\hat{\psi}_{\varepsilon r}}{Q(1-\hat{\psi})} + \left(\hat{G}\right)_{OLD} \tag{3.14}$$

where $\hat{\psi}_{\varepsilon r}$ is the estimated transfer function ε / r, $\left(\hat{G}\right)_{OLD}$ and $\left(\hat{G}\right)_{NEW}$ are the old and updated model, respectively. Thus with $\hat{\psi}_{\varepsilon r}$ the detection of MPM is immediate: if $\hat{\psi}_{\varepsilon r}$=*0*, then there is no MPM present, otherwise the old model may be updated using Equation (3.14).

3.5.1 Gain and Dead-time Correction

In this section, a simple iterative procedure is described which can detect gain and dead-time mismatch for SISO MPC independent of the model order or parameterization. If the gain is under-estimated then the controller will make overly aggressive control actions possibly leading to instability, and when it is over-estimated the control performance becomes sluggish. Closed-loop stability is especially sensitive to dead-time mismatch.

Result 1:

If Q, G, $\hat{G}$, and the closed-loop response $\psi = \varepsilon / r$ are all stable LTI systems, and $K_Q = 1/K_{\hat{G}}$ (for offset-free tracking), then

$$K_{\psi} = 1 - \frac{K_{\hat{G}}}{K_G} \tag{3.15}$$

$$K_{\psi_d} = (K_{\hat{G}} - \frac{K_{\hat{G}}^2}{K_G}) \tag{3.16}$$

where $K_{\psi_{\varepsilon r}}$, K_{ψ_d}, K_P, K_Q, and $K_{\hat{p}}$ are the gains of $\psi_{\varepsilon r} = \varepsilon / r$, $\psi_d = \varepsilon / u_d$, the process, controller, and model respectively.

Corollary 1:

Assuming that $K_G > 0$, and $K_{\hat{G}} > 0$ then,

$$K_G > K_{\hat{G}} \quad \Leftrightarrow \quad K_{\psi_{\varepsilon r}} > 0 \tag{3.17}$$

$$K_G < K_{\hat{G}} \quad \Leftrightarrow \quad K_{\psi_{\varepsilon r}} < 0 \tag{3.18}$$

$$K_G = K_{\hat{G}} \quad \Leftrightarrow \quad K_{\psi_{\varepsilon r}} = 0 \tag{3.19}$$

Thus, by identifying the gain of $\psi_{\varepsilon r}$ one can determine if the model gain is overestimated or underestimated. Appropriate corrections can be made to the gain, and $\psi_{\varepsilon r}$ re-identified to ensure the model gain is satisfactory. Appropriate results are easily derived for negative gains.

Theorem 2:

If $\hat{D} \leq D$ then the dead-time of $\psi_{\varepsilon r}$ is equal to $\hat{D}$. (3.20)

If $\hat{D} \geq D$ then the dead-time of $\psi_{\varepsilon r}$ is equal to D. (3.21)

Proof:

In Equation 1, if $\hat{D} \leq D$, then $\hat{D}$ may be factored out as:

$$\frac{\varepsilon}{r} = q^{-\hat{D}} Q(q^{-D+\hat{D}} G^* - \hat{G}^*) * \frac{1}{1 + Q(q^{-D} G - q^{-\hat{D}} \hat{G})}$$

where G^* represents the transfer function with the delay factored out. Since the controller Q and the second term on the RHS have no delay, then the dead-time of ε / r is $\hat{D}$. A similar reasoning may be applied in the case $\hat{D} \geq D$.

Theorem 2 may be used as a guide for adjusting the model dead-time. If ε / r has a dead-time equal to the model dead-time $\hat{D}$, then the model dead-time may be increased slightly until the delay of ε / r stops matching $\hat{D}$. If the delay of ε / r is not equal to $\hat{D}$, then this implies that the model dead-time is overestimated. In this case $\hat{D}$ may be reduced until the delay of ε / r starts to match $\hat{D}$. The results of Theorems 1 and 2 also apply for continuous systems.

The advantage of the above methods for correcting delay and gain are that they do not require any transformations of the closed-loop model ε / r to determine the mismatch. Thus, it does not require detailed knowledge of the controller, or any further processing. The gain and delay of ε / r are used directly to indicate mismatch, and to determine if the gain or delay is over/under estimated.

Transient analysis through 'bump testing' may be sufficient for identifying ε / r since this method can provide good estimates of the gain and dead-time. This approach relies on making a large enough perturbation such that the signal to noise ratio is large enough to allow easy identification of the process dynamics. The advantage of transient analysis is that is it simple, and intuitive. However, large perturbations may excite nonlinearities, and disturb the product quality. In these cases, one may use a correlation analysis to estimate the impulse response since it requires smaller perturbations. For the correlation analysis, it is important that the inputs are chosen to be statistically independent of the disturbance. The advantage of correlation analysis is that it is a nonparametric approach, which avoids the problems associated with model order structure.

The correlation analysis is now described. More detailed information about the correlation analysis may be obtained from [65]. If the model is assumed to be a finite impulse response (FIR) truncated after P intervals (which is reasonable for stable processes), then the response y(k) is related through the impulse response g(k) to the input u(k) and a disturbance v(k) as:

$$y(k) = \sum_{i=0}^{P-1} g(i)u(k-i) + v(k) \tag{3.22}$$

Assuming the series is ergodic and stationary, then the cross-correlation between the input and output $R_{yu}(\tau)$ may be calculated as:

$$R_{yu}(\tau) = E[y(k)u(k-\tau)] \tag{3.23}$$

$$R_{yu}(\tau) = \sum_{i=0}^{P-1} g(i)E[u(k-i)u(i-\tau)] + E[v(k)u(k-\tau)] \tag{3.24}$$

If the input is statistically independent of the disturbance, and has zero mean, then $E[v(k)u(k-\tau)] = 0$, and

$$R_{yu}(\tau) = \sum_{i=0}^{P-1} g(i)R_{uu}(\tau - i) \tag{3.25}$$

If the input is white noise with variance σ^2, then

$$\begin{aligned} R_{uu}(\tau - i) &= \sigma^2 \quad \tau = 0 \\ &= 0 \quad \tau \neq 0 \end{aligned} \tag{3.26}$$

which simplifies Equation(3.25). The impulse response coefficients may simply be calculated as:

$$g(i) = R_{yu}(k) / R_{uu}(0) \tag{3.27}$$

Example:

The above methods will be demonstrated using the following SISO process example from [88]:

$$G = \frac{(2s+1)e^{-8s}}{(10s+1)(7s+1)(3s+1)} \tag{3.28}$$

which is modeled with

$$\hat{G} = \frac{1.3*(2s+1)e^{-2s}}{(12s+1)(5s+1)(3s+1)} \tag{3.29}$$

It may be noted that the gain is overestimated, and the dead-time is underestimated. Mismatch also exists for two of the system poles. The following IMC controller system is obtained after inverting and adding a robustness filter:

$$Q = \frac{(12s+1)(5s+1)(3s+1)}{1.3*(2s+1)}\left(\frac{1}{10s+1}\right)^2 \tag{3.30}$$

The disturbance is modeled as the following transfer function

$$v = \frac{1}{4s+1}a \tag{3.31}$$

where a is white noise with a mean zero, and variance of 0.1. The disturbance is therefore coloured noise. The step responses of the process and the model (Equation (3.32)) are shown in Figure 3-7. The model is mismatched, and in particular the model has an overestimated gain mismatch, and an underestimated dead-time. For the identification of $\psi_{\varepsilon r} = \varepsilon / r$, a white noise dithering signal of zero mean and variance 0.5 was applied to the set-point r (Figure 3-8). Data was collected using a sampling rate of 0.5, Using

MATLAB's system identification toolbox, a correlation analysis was performed resulting in an estimated step response for $\psi_{\varepsilon r}$. The estimates of $K_{\psi_{\varepsilon r}}$ and the dead-time of $\psi_{\varepsilon r}$ are –0.31 and around 4-5 respectively. From Equation (3.18) it is known that the gain is overestimated, and using Equation (3.15) the process gain is estimated as 1.3/1.31=0.99. The model and controller are updated to account for the improved gain estimate. The estimated dead-time of $\psi_{\varepsilon r}$ is 2-3, which is close to $\hat{D}$ (note that since the sampling time is 0.5 the time must be divided by 2 in Figure 3-9). From Result 2, this implies that $\hat{D}$ is too small. $\hat{D}$ is increased to 4, and the resulting impulse response estimate of $\psi_{\varepsilon r}$ is shown in Figure 3-10. The 99% confidence intervals estimated via MATLAB are also shown to aid in the determination of the delay. It is not critical to exactly estimate the delay, but to determine if the delay of $\psi_{\varepsilon r}$ is increasing with $\hat{D}$, which would indicate that the model delay is still underestimated. Since the delay has increased from about 2-3 to about 5 we would determine that the delay is still underestimated. When the model delay is increased to 6 (Figure 3-11) the estimated delay of $\psi_{\varepsilon r}$ is also seen to increase to about 6-8. Again this indicates that the delay is still underestimated. When the delay was changed to 8, none of the impulse response coefficients were statistically different from zero (Figure 3-12). The model error is so small at this point that it cannot be resolved unless more data are collected or the signal to noise ratio is increased. This would indicate that the delay is not correctly estimated. If the delay is further increased to greater than 8 it would be observed that the delay of $\psi_{\varepsilon r}$ is no longer a function of the model delay, which also indicates that 8 is the correct delay. Theorem 2 implies that the dead-time of the model should be equal to 8. A comparison of the step responses of the original model and the improved model is shown in Figure 3-13.

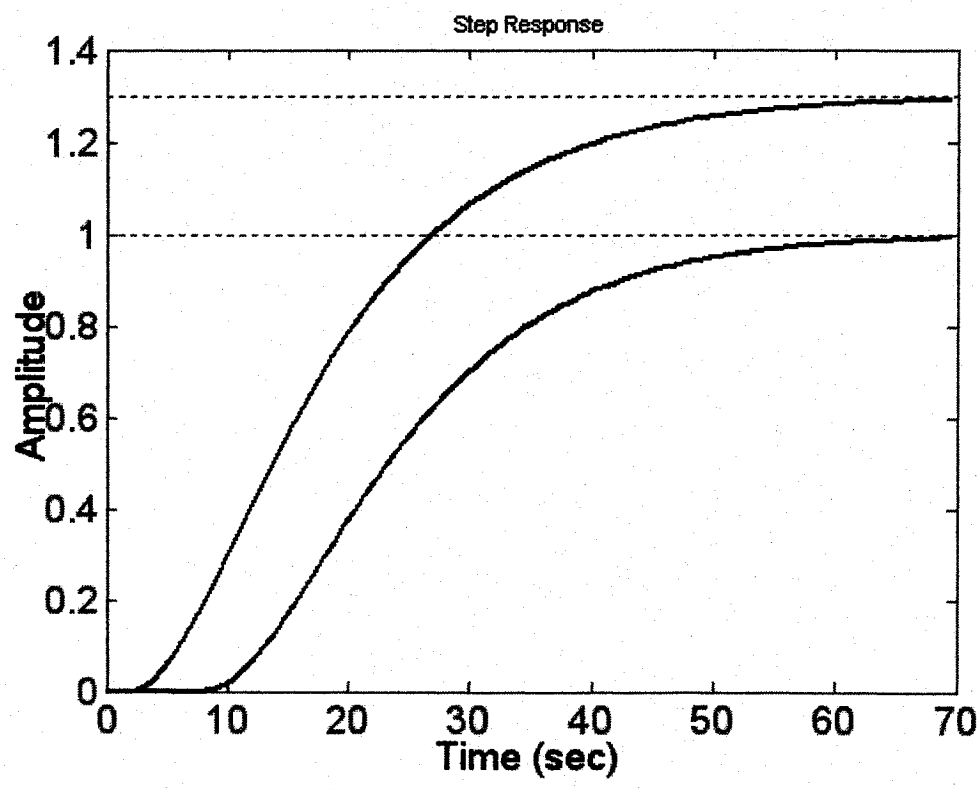

Figure 3-7. Step response of process and model.

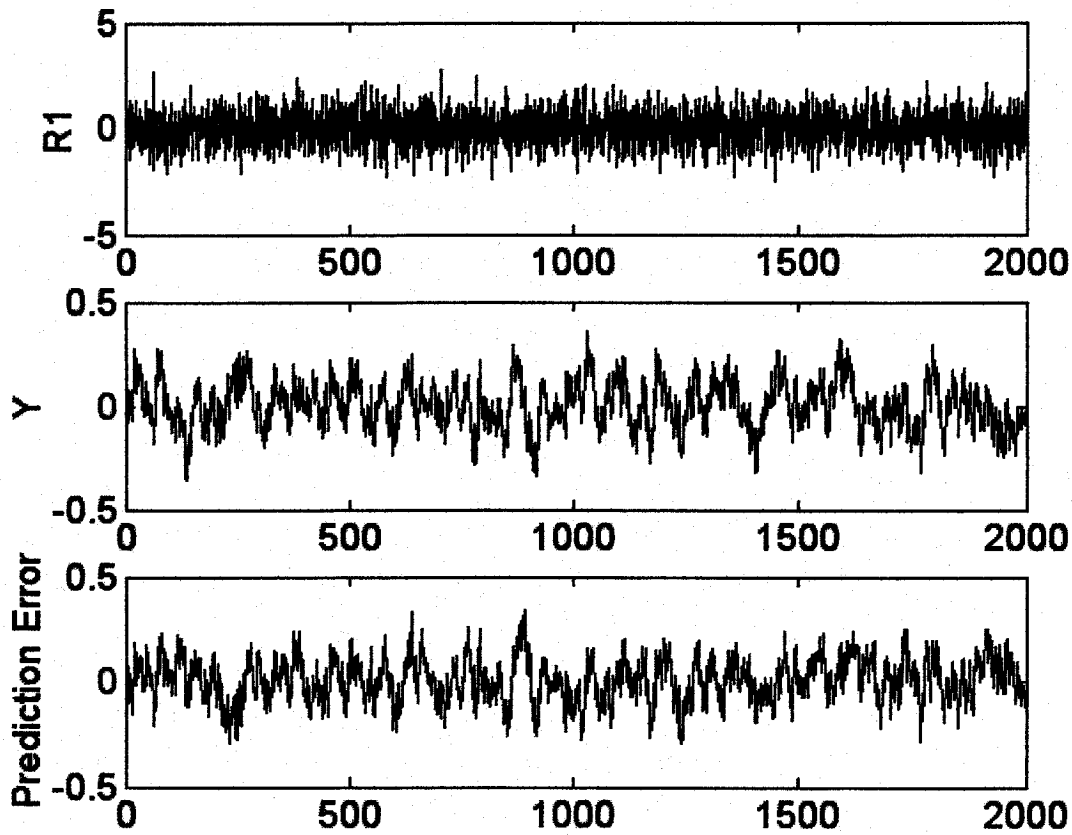

Figure 3-8. White noise applied to set-point *r* and the resulting output *y* and prediction error *e*.

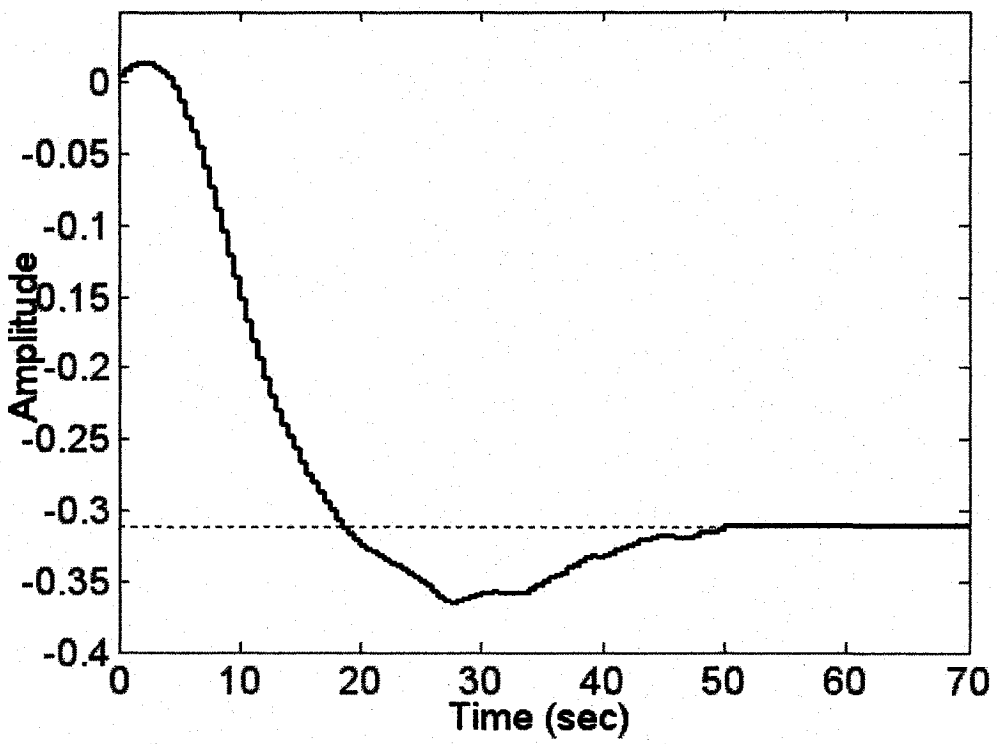

Figure 3-9. Estimated ψ_{er}.

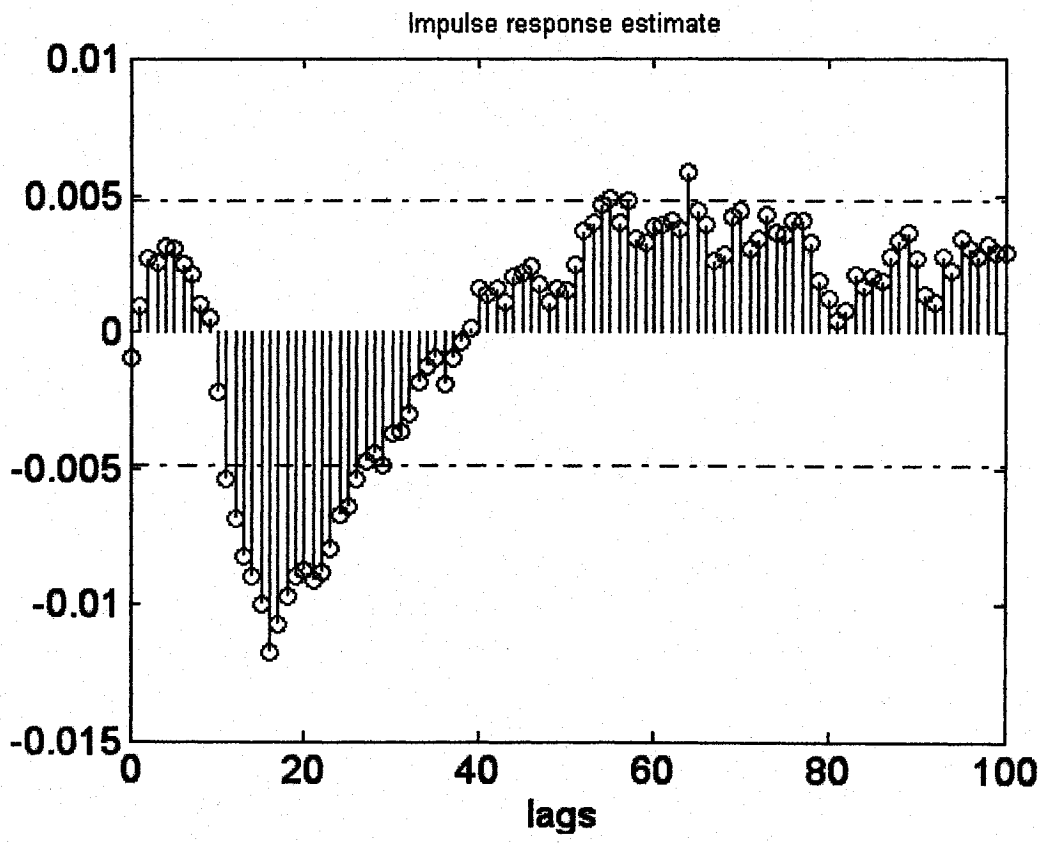

Figure 3-10. Estimated $\psi_{\varepsilon r}$ with improved gain and delay estimate of 4.

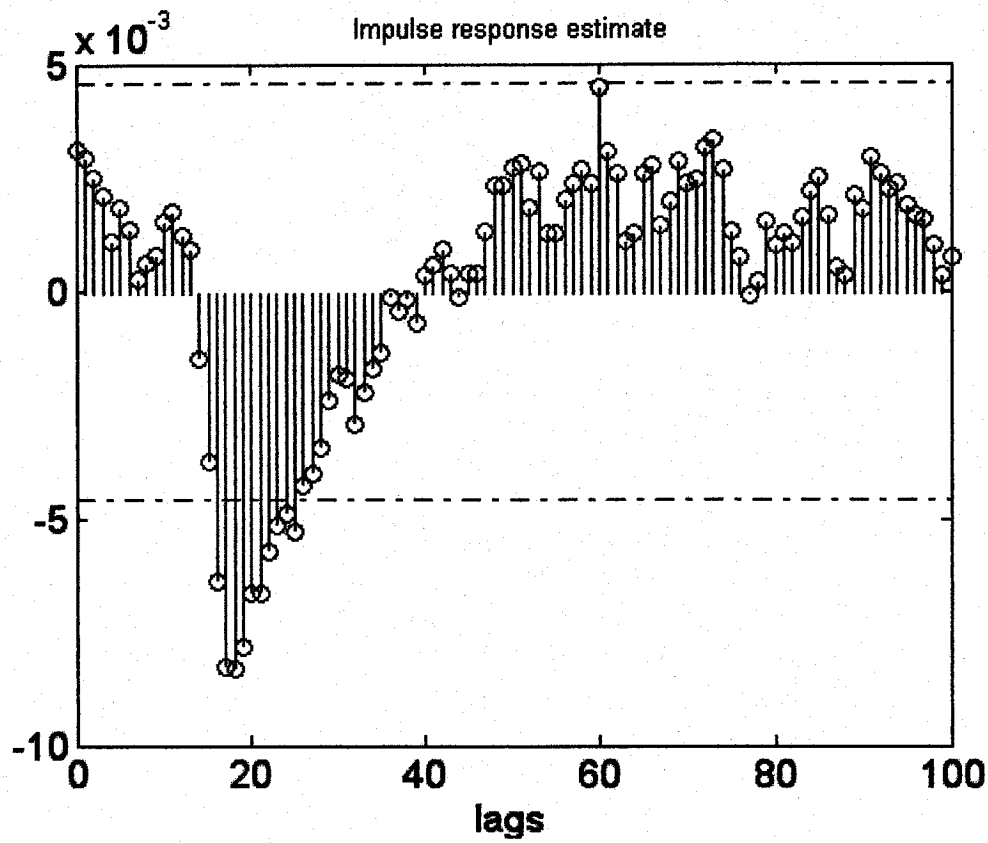

Figure 3-11. Estimated ψ_{er} with improved gain and delay estimate of 6.

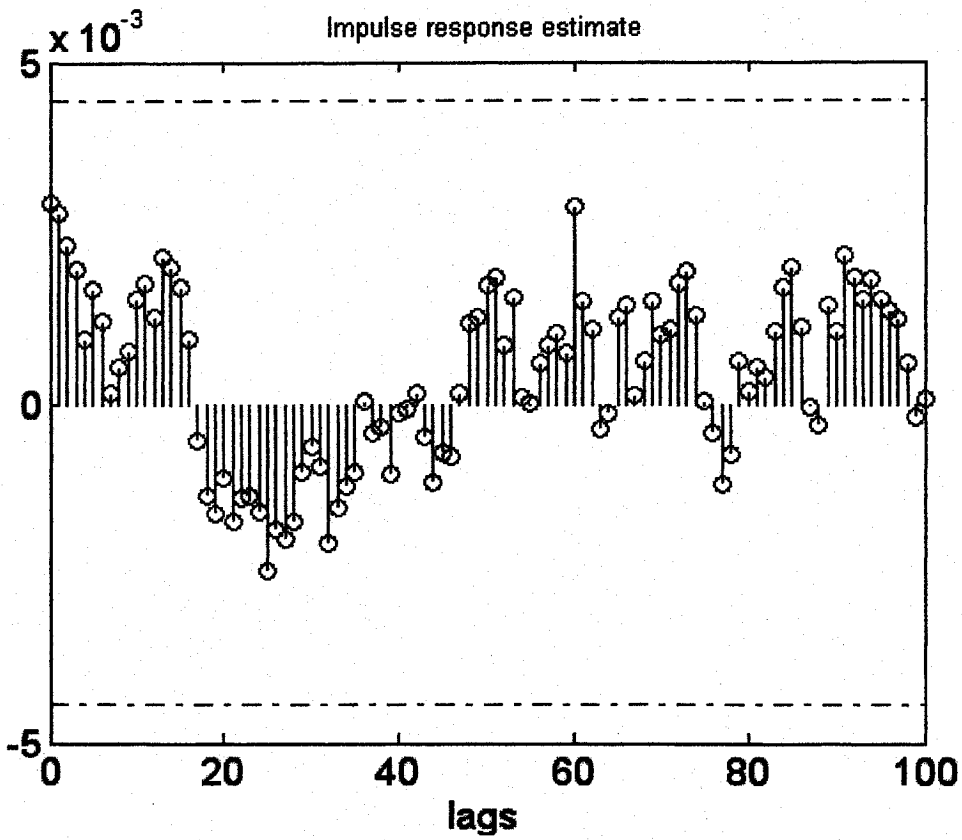

Figure 3-12. Estimated ψ_{sr} with improved gain and delay estimate of 8.

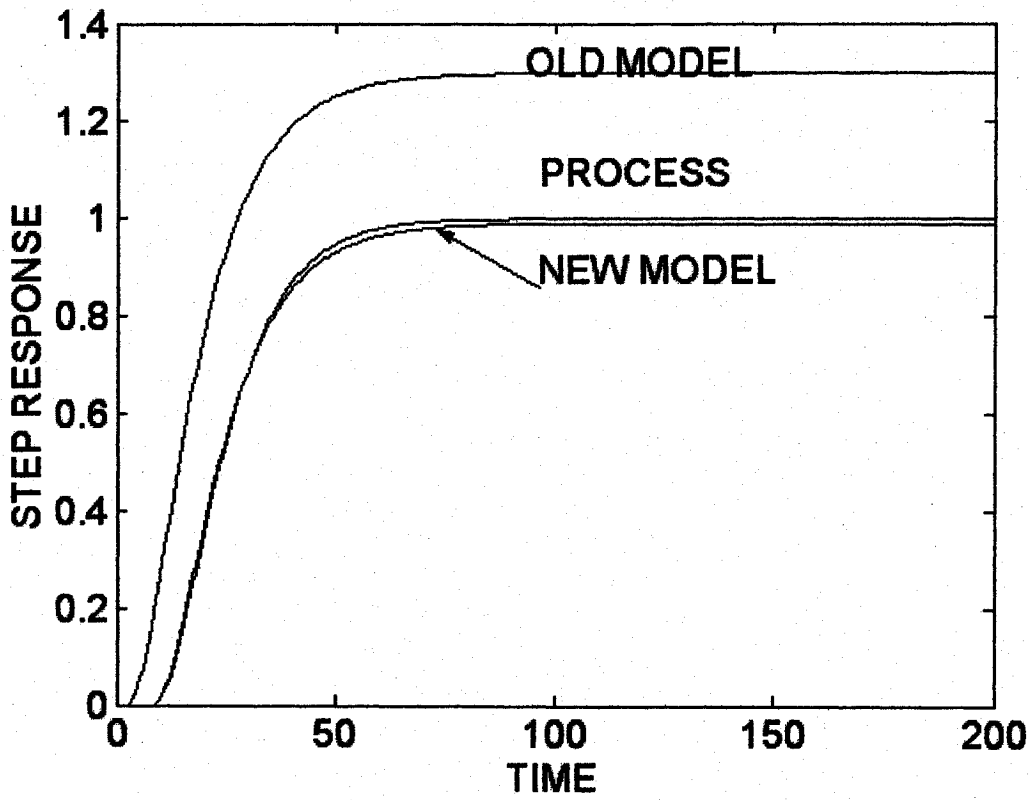

Figure 3-13. Comparison of old and new models.

A method has been presented to detect and correct for gain and dead-time mismatch using closed-loop data. The identification of the model between the set-point and prediction error is effectively open loop, which allows the use of standard identification techniques such as transient analysis or correlation analysis. The nonparametric methods allow for good estimates of the dead-time and gain of $\psi_{\varepsilon r}$, which can then be used to update the process model's dead-time and gain. To correct for other forms of mismatch some parametric methods of identification can be used, and with knowledge of the process model and controller, an estimate for the process may be calculated.

3.5.2 Manual online update of first order + dead time model parameters

This section examines if the results of the previous section can be used to manually update a first-order plus dead time (FOPDT) approximation to a higher order system. A FOPDT model has three parameters: the gain, dead time and time constant. These models are used to approximate high order over-damped systems, which commonly occur in the chemical and petroleum industry. The basic idea is that if the gain and delay can be suitably corrected then any remaining mismatch should be a result of time constant mismatch. Assuming that the gain, and dead time have been previously corrected using the above methods, then any pattern similar to Figure 3-4 in the prediction error is a result of time constant mismatch. The time constant may be adjusted until there is no significant pattern indicating time constant mismatch left in the prediction error. This would involve an iterative process in which the time constant is increased or decreased according to the information in the prediction error pattern. Note that the controller is redesigned with each change in time constant. When the pattern flips its sign, then it is an indication that the time constant was increased or decreased too far. We consider an

example where the true process is not a FOPDT, but a third order over-damped process given by:

$$G = \frac{e^{-3s}}{3s^3 + 9s^2 + 5s + 1} \tag{3.33}$$

The open-loop step response of Equation (3.33) is shown in Figure 3-14. Visual inspection reveals that a delay of 3, gain of 1, and a time constant of 5 would be appropriate for approximating the system via a FOPDT model.

The following controller and initial model were used:

$$Q = \frac{\hat{\tau}s + 1}{\hat{K}(15s + 1)} \qquad \hat{G} = \frac{0.7e^{-2s}}{10s + 1} \tag{3.34}$$

The output response to a unit set point change is shown in Figure 3-15. From Figure 3-15 the delay is estimated to be 3, and hence from Equation (3.21) the model dead time should be changed from 2 to 3. The prediction error response is shown in Figure 3-16, and displays a positive gain, which from Result 1 indicates that the gain is underestimated. The model dead time is changed to 3, the gain is increased slightly, the controller is redesigned, and the test is repeated. These steps are repeated by making small adjustments to the model gain until the apparent gain of the prediction error is zero.

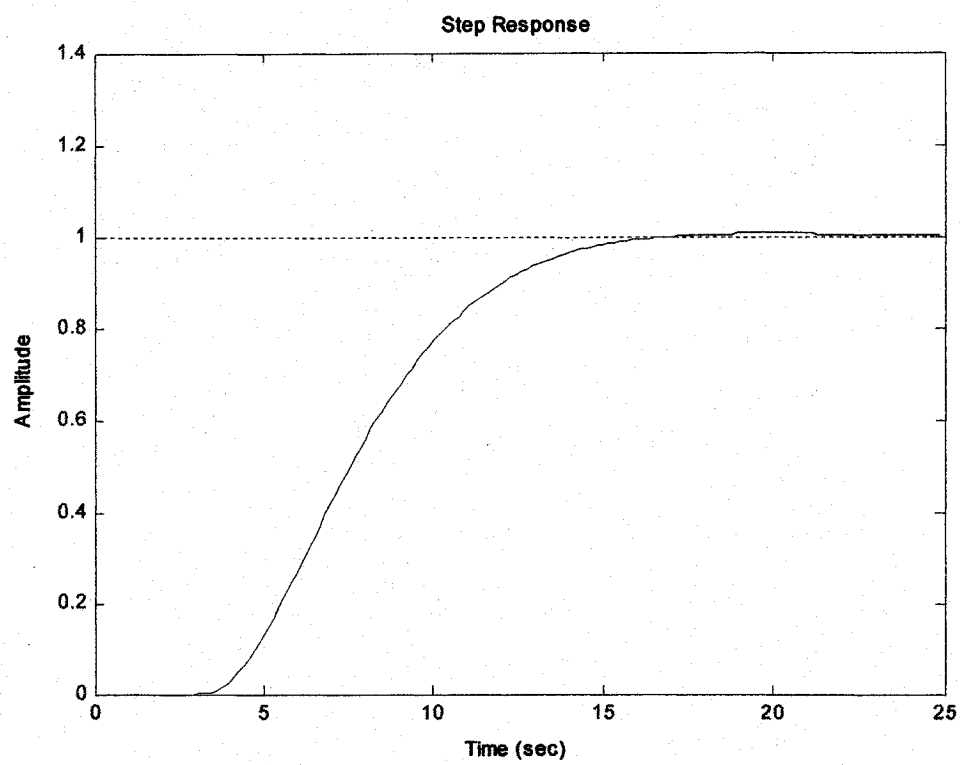

Figure 3-14. Step response of process given by Equation (3.33).

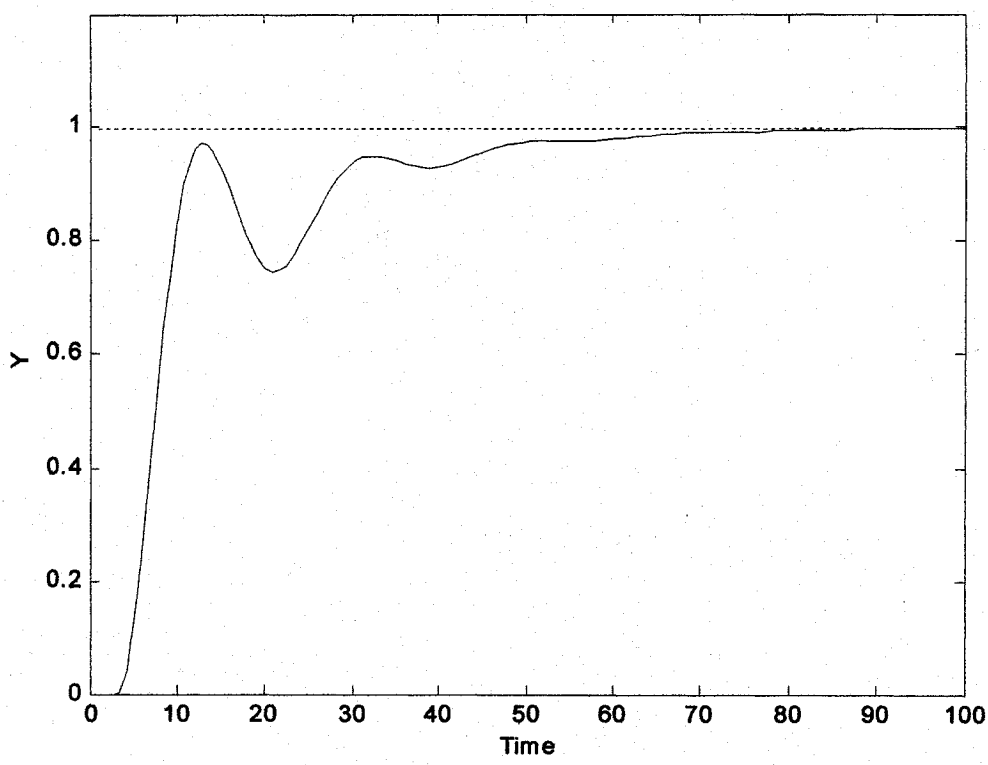

Figure 3-15. Response of output to unit set point change.

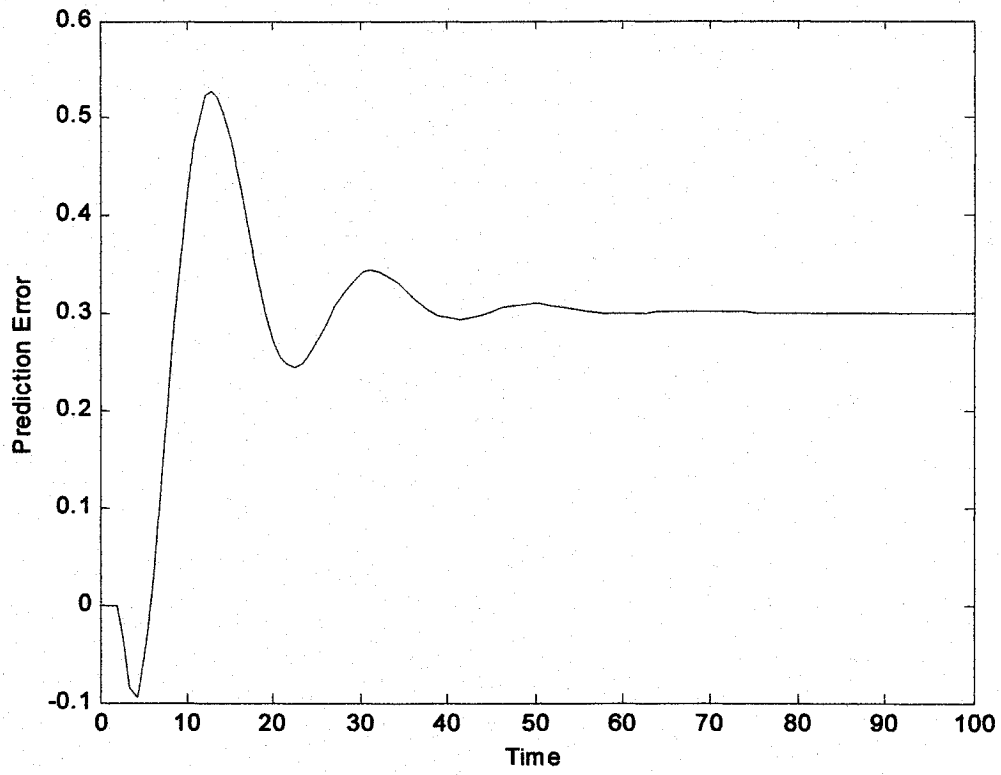

Figure 3-16. Response of prediction error to a unit set point change.

The gain of the prediction error appeared to be zero when the model gain was increased to 1.0 (Figure 3-17). Ignoring the small negative 'blip' at t = 4, the main pattern in Figure 3-17 indicates that the time constant is overestimated. The model time constant is reduced from 10 to 7, and the new prediction error pattern is shown in Figure 3-17. The peak at t = 15 is reduced from 0.24 to 0.15, which indicates an improvement, however there is still an indication that the time constant is mismatched. Therefore the time constant is further reduced to 5. Now the pattern indicates no time constant mismatch. If the time constant had been reduced too far, say to 2, then the peak would flip to the opposite side indicating the time constant has been reduced too much.

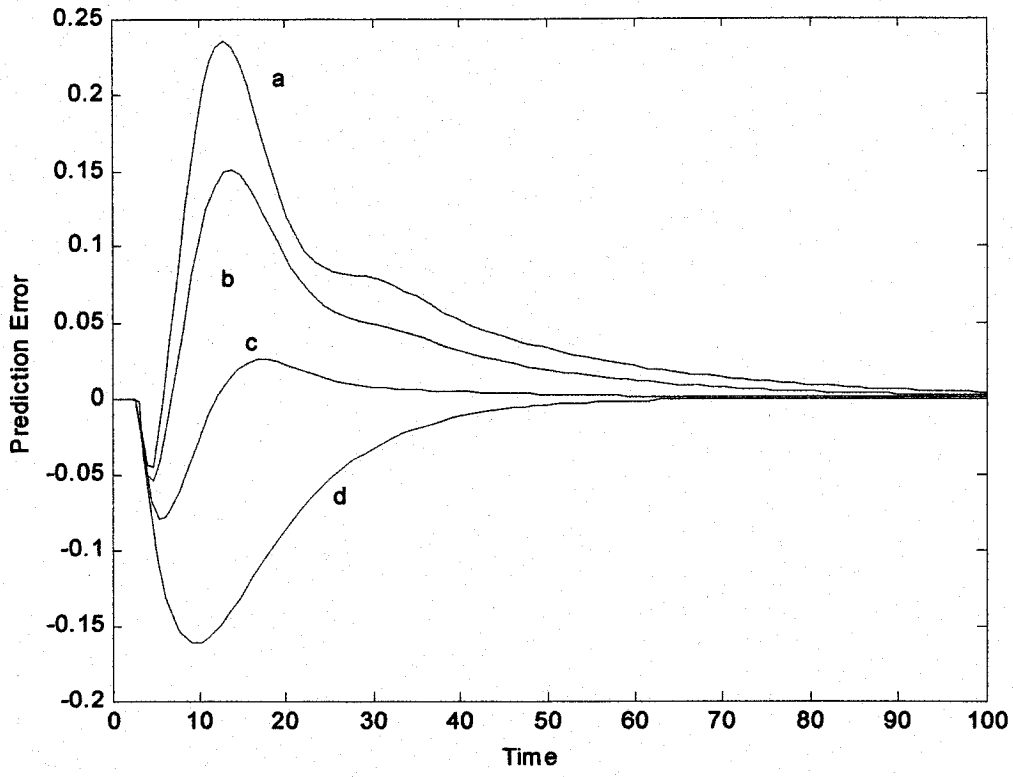

Figure 3-17. Prediction error patterns after the gain and dead time have been corrected. a: $\hat{K}=1,\hat{\tau}=10,\hat{D}=3$, b: $\hat{K}=1,\hat{\tau}=8,\hat{D}=3$, c: $\hat{K}=1,\hat{\tau}=5,\hat{D}=3$, $\hat{K}=1,\hat{\tau}=2,\hat{D}=3$

Thus the new model is $\hat{K}=1, \hat{\tau}=5, \hat{D}=3$ which corresponds well with what is expected from Figure 3-14. The model's step response before and after the updating is shown in Figure 3-18. This method provides simple visual indicators that may be used to adjust a FOPDT model online

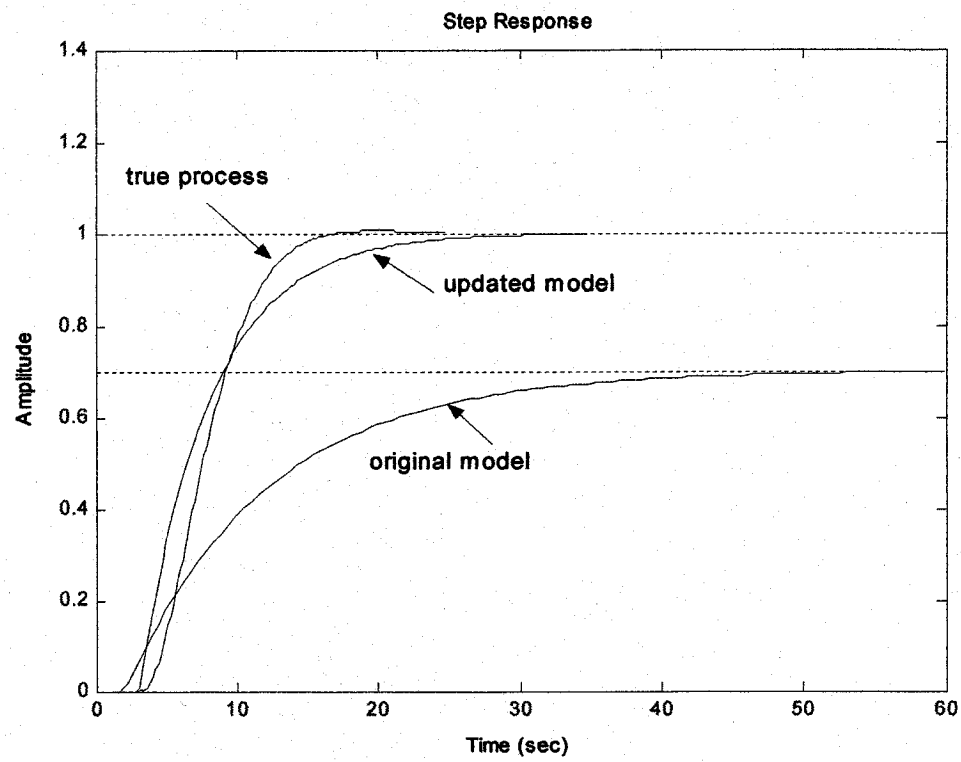

Figure 3-18. Comparison of step responses of the models before and after to the true system.

3.6 Multivariate systems

This section considers the diagnosing of MPM in multivariable systems operating under closed-loop conditions and determining the subset of models that need re-identification. Consider, for example, Dynamic Matrix Control (DMC) [78], which utilizes one step response model for each input-output pair. DMC with ten inputs and ten outputs would have 100 (10×10 = 100) step response models. We wish to determine which specific subset of input-output pairs should be candidates for re-identification. Accurately re-identifying all of the process models is an expensive, intrusive, and time-consuming effort, which requires a relatively high degree of specialized knowledge. A large part of the difficulty lies in the proper selection of the model structures for the development of parsimonious parametric models. Hence, it would be useful to have simple preliminary screening tools that can provide information as to which specific subset of models should be considered for re-identification. Ideally, the methods should be minimally intrusive, and allow for the process to remain under full or partial control. The goal of this section is to present methods to determine which elements $\hat{G}_{ij}$ in the model transfer function matrix, $\hat{G}$ are mismatched and should be considered for re-identification. $\hat{G}_{ij}$ refers to the transfer function between output i and input j.

3.6.1 Detecting rows and columns with MPM

Stanfelj used the cross-correlation between r (or u_d) and ε to distinguish between disturbances and model error [9], However as shown in this section, for MIMO systems this approach must be modified if it is to help pinpoint which particular models are contributing to the MPM. This section examines what happens to the covariance or cross-correlation between the external dithering signal u_d and the prediction errors when a row or column of $\hat{G}$ does not contain any MPM. The purpose is to determine which columns and rows in $\hat{G}$ contain MPM. Then we can infer which subset of models should be candidates for re-identification. Consider a 3×3 model transfer function matrix shown in Figure 3-19. If only a single row and column are detected as containing mismatch (Figure 3-19(a)) then it is unambiguous as to which model contains the mismatch, in this example $\hat{G}_{33}$. If for example, we identify Rows 1 and 3 and Columns 1 and 3 as containing MPM (Figure 3-19(b)), this would indicate that models $\hat{G}_{11}$, $\hat{G}_{13}$, $\hat{G}_{31}$, $\hat{G}_{33}$ may be mismatched.

(a) (b)

Figure 3-19: Examples of row and columns containing mismatch in a 3×3 model transfer function matrix.

Proof.

Using the so-called 'push-through' rule $A(I+BA)^{-1} = (I+AB)^{-1}A$, and setting $r = 0$

$$\varepsilon = (I+\theta Q)^{-1}\theta\, u_d + (I+\theta Q)^{-1} v = \theta(I+Q\theta)^{-1}u_d + (I+\theta Q)^{-1} v$$

Now, if $row_i\theta = 0$ then $row_i\left(\theta(I+Q\theta)^{-1}\right) = 0$

Therefore,

$$\varepsilon_i = \sum_{j=1}^{n} u_{d_j} \bullet 0 + \sum_{j=1}^{n} v_j \bullet \left[(I+\theta Q)^{-1}\right]_{ij}$$

$$\varepsilon_i u_{d_k} = \sum_{j=1}^{n} u_{d_k} v_j \left[(I+\theta Q)^{-1}\right]_{ij}$$

Taking expectations on both sides yields (if u_{d_k} is independent of the disturbance v):

$$E(\varepsilon_i u_{d_k}) = 0 \quad \forall k = 1,2,..n$$ ▲▲▲

RESULT 2:

If $col_i\theta = 0$ then the covariance at all lags between u_{di} and each of the $\varepsilon_i \forall j = 1,2,...,n$ will be zero.

The proof follows similar lines as in the above proof and is not shown.

Therefore using Results 1 and 2, the rows and columns of $\hat{G}$ which contain MPM may be identified by examining the cross correlation at different lags between u_d and the ε. An advantage of this method is that no assumptions are made concerning the disturbance, other than it is independent of the dithering signal. This will make the tests

more robust against changing disturbance dynamics. The procedure is demonstrated through the following examples.

Example 1:

Consider the Shell oil fractionator 3×3 system adapted from [76]:

$$G=\begin{pmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.72e^{-14s}}{60s+1} & \frac{6.90e^{-15s}}{40s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.2}{19s+1} \end{pmatrix} \tag{3.39}$$

A stochastic disturbance is modeled as:

$$v=G_d a=diag(\frac{1}{45s+1},\frac{1}{25s+1},\frac{1}{19s+1})a \tag{3.40}$$

where a is a 3×1 column vector of mean zero white noise, and covariance matrix $\Sigma_a = diag(0.001, 0.001, 0.001)$, G_d is the disturbance transfer function. A zero-order hold with a sampling time of T = 10 is used to discretize the system. An unconstrained DMC controller with a prediction horizon of 50 and control horizon of 3 intervals is used to control the system. The diagonal elements of the dynamic matrix are multiplied by a factor of 1.01 for move suppression. The DMC controller is designed using the nominal system given by Eq. (4). However, now consider that the process G_{21} changes:

$$G_{21} = \frac{5.39e^{-18s}}{50s+1} \Rightarrow \frac{6.1e^{-21s}}{45s+1} \tag{3.41}$$

Thus the model $\hat{G}_{21}$ now contains MPM. A mean zero white noise signal with covariance matrix $\Sigma_a = diag(1.0,1.0,1.0)$ is applied to u_d for 2000 samples. If desired each component of the vector u_d may be excited one at a time, resulting in a larger number of experiments. A smaller dithering signal may used if balanced by collecting more points.

The sample cross-covariances between ε_i and u_{dj} at lag k are calculated as:

$$C_{\varepsilon_i - u_{d_j}}(k) = \frac{1}{N}\sum_{t=1-k}^{N-k}(\varepsilon_i(t+k) - \bar{\varepsilon}_i)(u_{d_j}(t) - \bar{u}_{d_j}) \quad k = 0,1,...N-1 \tag{3.42}$$

where N is the number of samples collected, and $\bar{\varepsilon}_i$ and $\bar{u}_{d_j}$ are the average values of ε_i and u_{d_j} respectively. The cross-covariance may be normalized by the individual variances to become the sample cross-correlation:

$$r_{\varepsilon_i - u_{d_j}}(k) = \frac{C_{\varepsilon_i - u_{d_j}}(k)}{\sqrt{C_{\varepsilon_i - \varepsilon_i}(0)}\sqrt{C_{u_{d_j} - u_{d_j}}(0)}} \tag{3.43}$$

The sample cross-correlations between ε and u_d and 95% confidence intervals are shown in Figure 3-20. Only Rows 2 and Column 1 show any cross-correlation significantly different from zero, indicating that model $\hat{G}_{21}$ is mismatched.

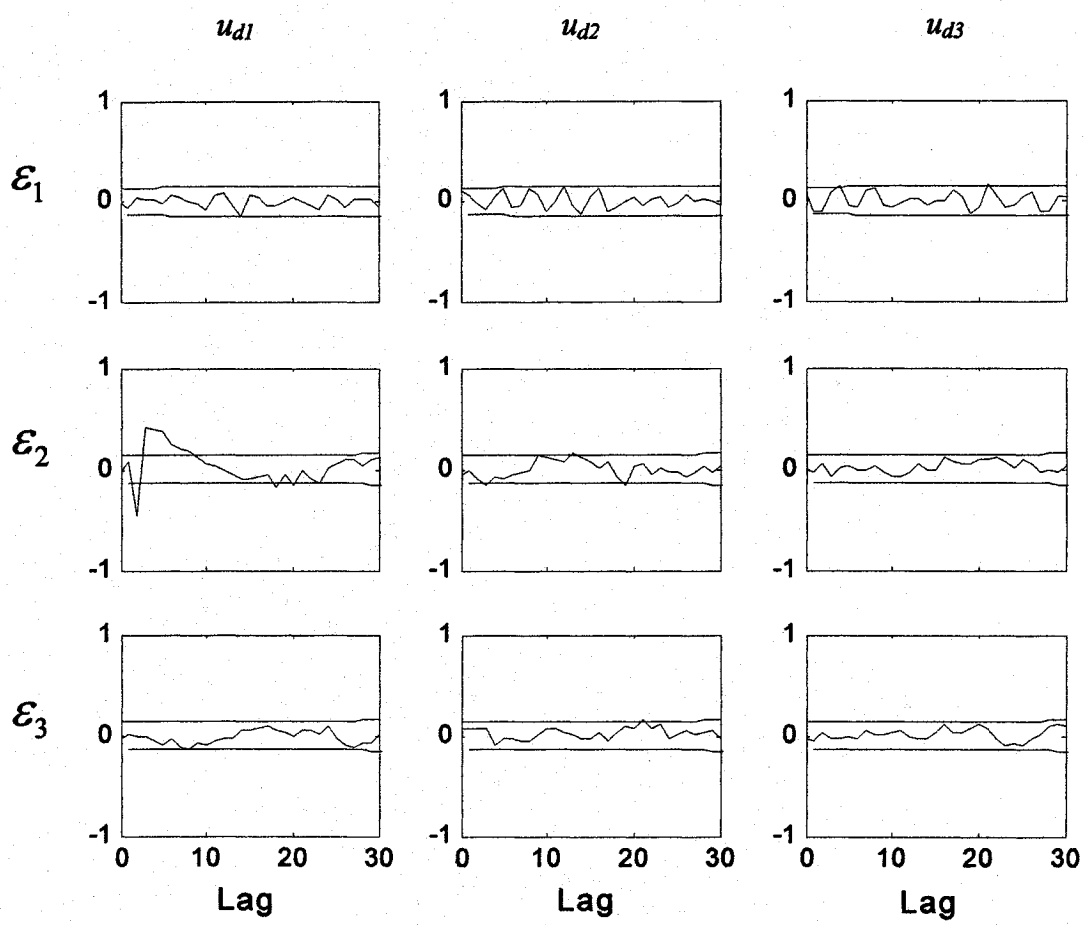

Figure 3-20. Cross-correlations at various lags between ε and dithering signal u_d for example 1. The 95% confidence intervals are also shown.

Example 2.

This example considers that the following three models, $\hat{G}_{21}, \hat{G}_{22}, \hat{G}_{31}$ are mismatched:

$$G_{21} = \frac{5.39e^{-18s}}{50s+1} \Rightarrow \frac{6.1e^{-21s}}{45s+1} \tag{3.44}$$

$$G_{22} = \frac{5.72e^{-14s}}{60s+1} \Rightarrow \frac{5.72e^{-14s}}{50s+1} \tag{3.45}$$

$$G_{31} = \frac{4.38e^{-20s}}{33s+1} \Rightarrow \frac{6.1e^{-30s}}{45s+1} \tag{3.46}$$

A similar experiment was run as in Example 1. The sample cross-correlations are shown in Figure 3-21. Since there are significant correlations for Rows 2 and 3, and Columns 1 and 2, there are four models that may have mismatch, namely $\hat{G}_{21}$, $\hat{G}_{22}$, $\hat{G}_{31}$, $\hat{G}_{32}$. These four models may be flagged for re-identification, or we may try to further reduce this set of candidate models by operating under partial control where a subset of the manipulated variables are held constant during the testing procedure described in the next section

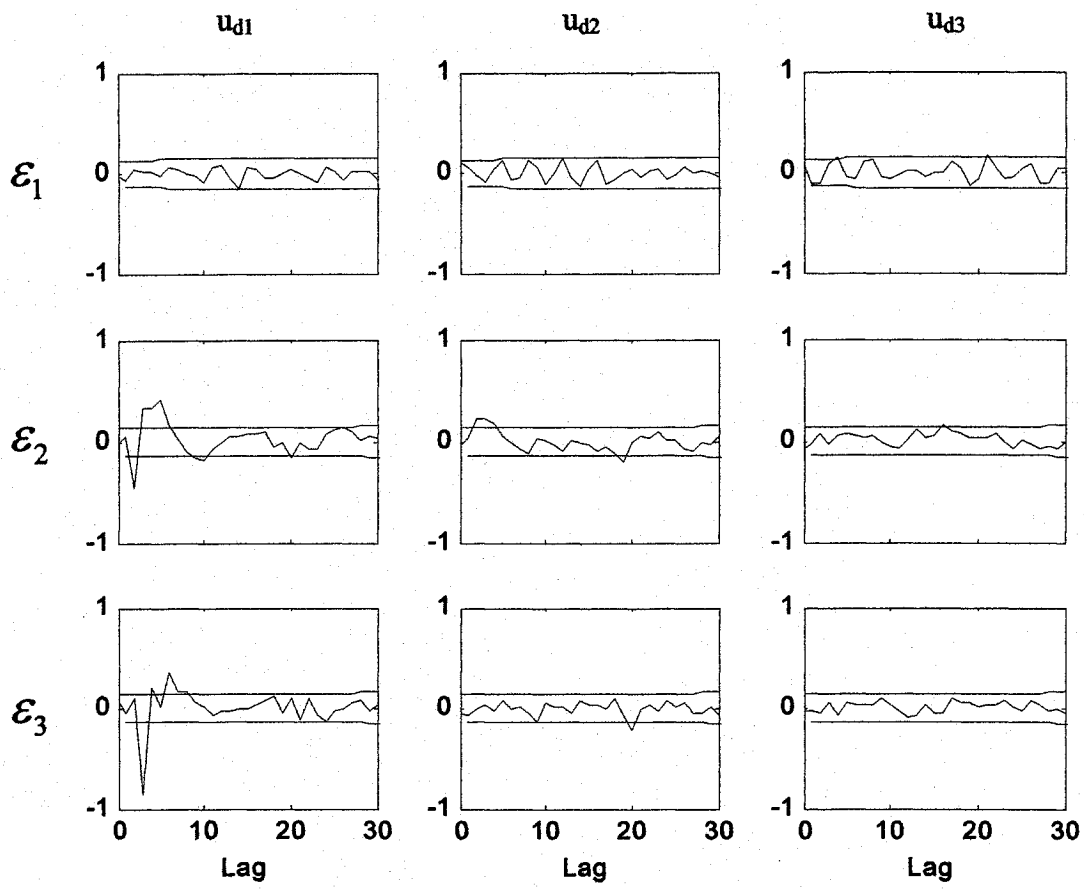

Figure 3-21. Cross-correlations at various lags between ε and dithering signal u_d for example 2. The 95% confidence intervals are also shown.

3.6.2 MPM detection under partial control

Our previous work [13]examined holding one input constant at a time to help determine which model is mismatched. The original method utilized a doublet (step up/hold/step down) set point trajectory, and visual examination of the resulting transient response of the preditition error. The strategy was that if a particular input that corresponds to a mismatched model is held constant, then the prediction error should exhibit no pattern. In this thesis, this approach is extended by using a correlation analysis. Moreover, it is shown that it may be necessary to hold subsets of the inputs constant instead of holding a single input constant when several models are mismatched. The proposed method has the following advantages:

- The process is not disturbed as much since the input is a small amplitude dithering signal applied to the setpoint or input signal. The transient analysis approach must implement a large enough step change in the setpoint for the pattern to be visible in the presence of noise. Futher, the large step can excite nonlinearites, which is avoided in the correlation analysis.

- The estimates are more robust to disturbances. The original pattern may be hidden in noise, unless a large signal to noise ration is used, which may upset the plant, and excite nonlinearites. The original method has to be repeated several times to average out the effect of disturbances.

- Instead of visual inspection of the patterns, the correlation between input dithering signal and prediction error is used which is less subjective.

The following 3×3 MIMO example problem is considered to illustrate the detection of imperfect models:

$$\begin{bmatrix} y_1(s) \\ y_2(s) \\ y_3(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{10s+1}e^{-3s} & \frac{4}{12s+1}e^{-5s} & \frac{-3}{11s+1}e^{-3s} \\ \frac{2}{10s+1}e^{-2s} & \frac{-3.6}{8s+1}e^{-4s} & \frac{3}{9s+1}e^{-3s} \\ \frac{-2}{12s+1}e^{-2s} & \frac{1}{14s+1}e^{-s} & \frac{-3}{8s+1}e^{-2s} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \\ u_3(s) \end{bmatrix} + \begin{bmatrix} v_1(s) \\ v_2(s) \\ v_3(s) \end{bmatrix} \quad (3.47)$$

An initial steady state: u_{1ss}=1, u_{2ss}=1 and u_{3ss}=1, is used. The adjustments are normalized with respect to the corresponding steady state values of output, y (y_{1ss}=2, y_{2ss}=1.4, y_{3ss}=-4). The measurement noise $v_i(s)$ is considered to be normally distributed with mean zero and a standard deviation of 1% of y_{iss}.

The original procedure in [13] to detect the input-output pairs that contain mismatched parameters consists of applying doublet changes simultaneously to the set-points of all the controlled variables. Then one can see which adjustment term (i.e. bias) is activated, that is, becomes non zero. If the i^{th} adjustment term (for the i^{th} output) is activated, then it means the mismatch is in the i^{th} row of the process transfer function matrix. Once the row(s) containing the mismatch are identified, the doublet changes in the set-points are repeated by holding one of the manipulated variable constant (at its initial steady state) at a time over the duration of the doublet change. If the mismatch is in the j^{th} column of the matrix in Equation (3.47), then the corresponding activity in the adjustment term will disappear when the j^{th} manipulated variable is held constant. The doublet should be of long enough duration such that errors due to gain mismatch have enough time to reveal themselves.

It may be noted that the magnitude of the bias term (adjustment term) depends on the degree of mismatch and the amount of movement in the corresponding manipulated variable. The change in the variance of the bias term over the duration of the doublet can provide additional clues in detecting the mismatched pairs. If the mismatch is in the j^{th} column, then the variance in the adjustment term will get reduced when the j^{th}

manipulated variable is held constant. The above discussion has assumed that there is only one mismatched model. If the mismatch occurs in more than one column of the same row, then more than one manipulated variable will need to be held constant at a time. Now the detection procedure is tested by considering the following two cases:

Case 1: Let us assume that the $y_1(s)/u_1(s)$ model contains a 30% gain mismatch, *i.e.*, $K_{model} = 1.3 \times K_{process}$. A doublet change in all three set-points is made at time =10. The responses of the output variables and normalized adjustment variables are shown in Figure 3-22 and Figure 3-23, respectively. Figure 3-23(a) shows that the adjustment for y_1 displays a pattern consistent with gain mismatch. The adjustment terms for the other two outputs show very minor activity due to the disturbance. Next, the same doublets are applied in the set-points, but u_1, u_2, and u_3, are held constant one at a time over the duration of the set-point changes (Figure 3-24). Figure 3-24(a) shows that the pattern previously observed for the adjustment for y_1 is absent when u_1 is held constant. The normalized variance is reduced to 0.0024 from 0.162. Figure 3-24(b) and (c) show the bias terms for y_1 for the same changes in set-points when u_2 and u_3, respectively, are held constant. In these cases the pattern observed in Figure 3-23(a) reappears pointing to a gain mismatch in the $y_1(s)/u_1(s)$ model.

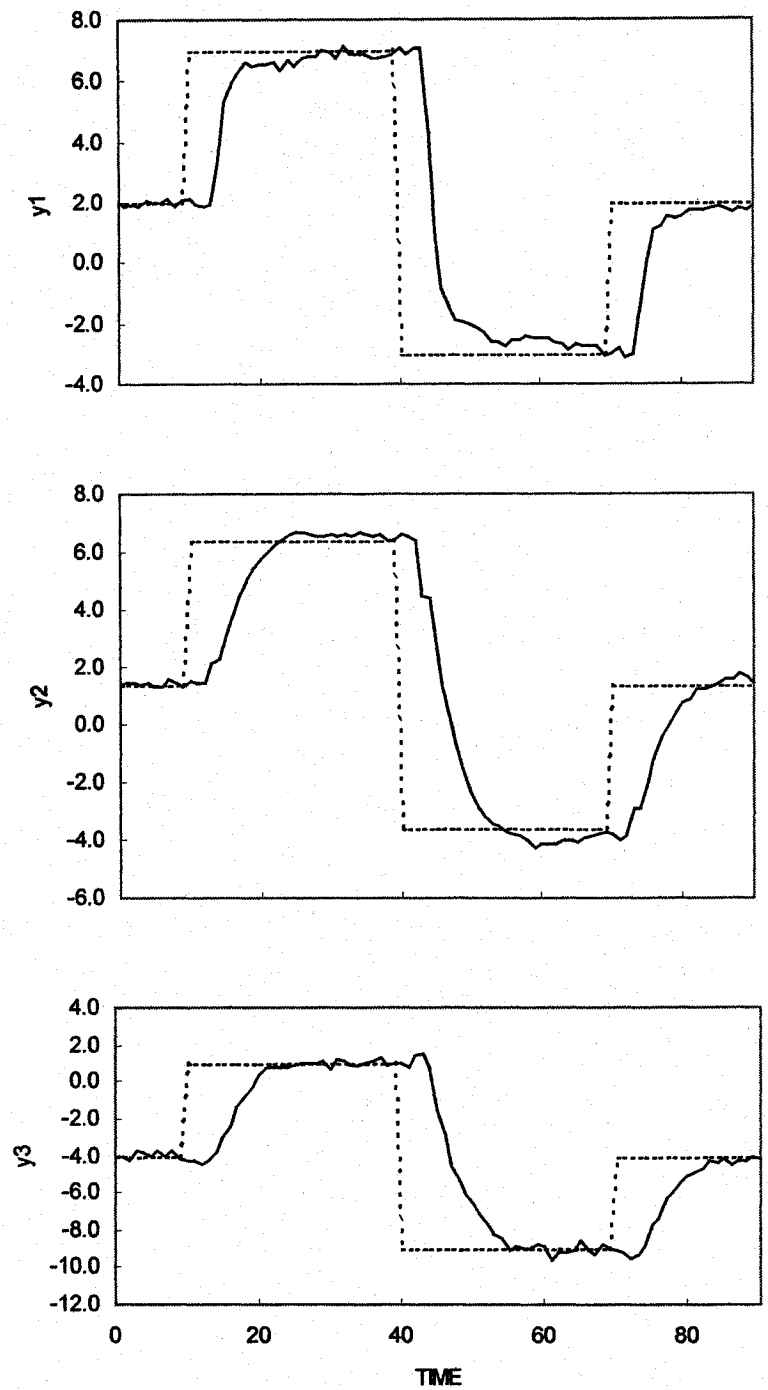

Figure 3-22. Response of the outputs to doublets changes in set-points when $y_1(s)/u_1(s)$ model contains 30% gain mismatch.

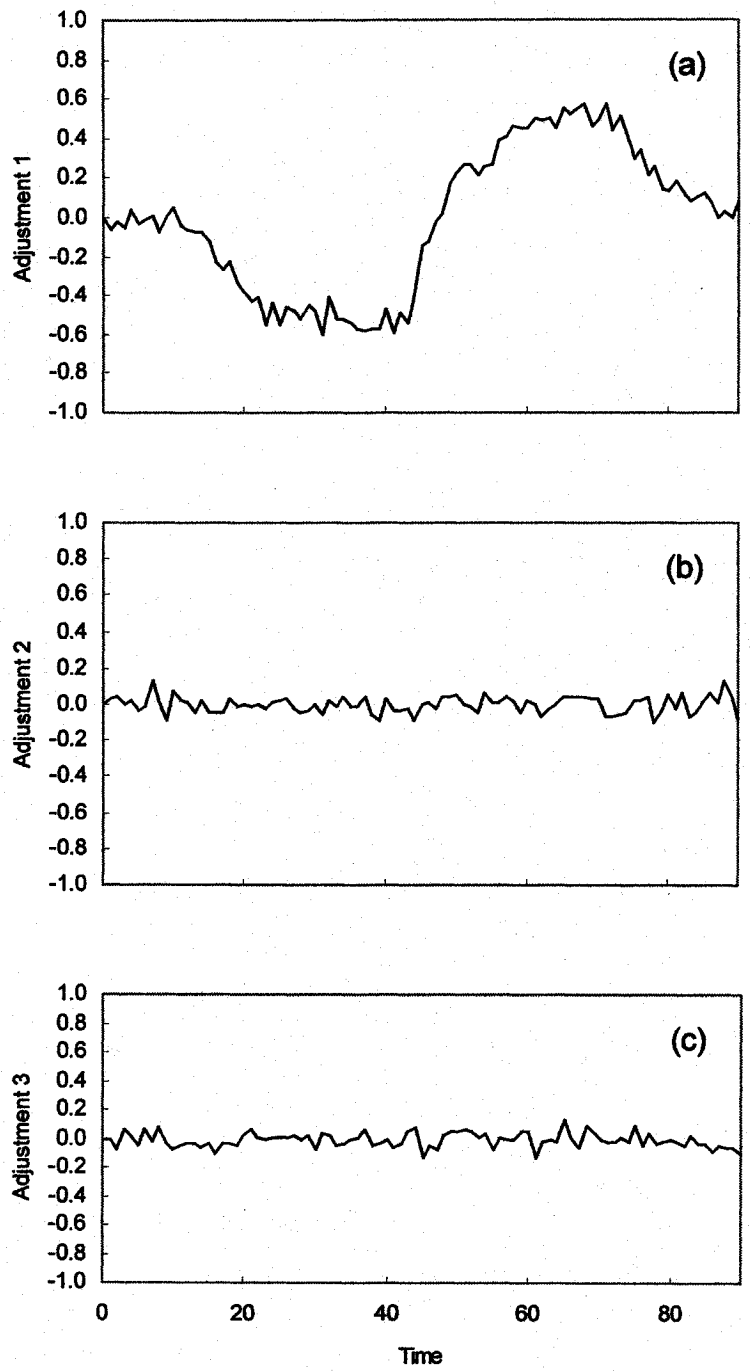

Figure 3-23. Response of the adjustments in the three outputs to doublet changes in set-points when $y_1(s)/u_1(s)$ model contains 30% gain mismatch.

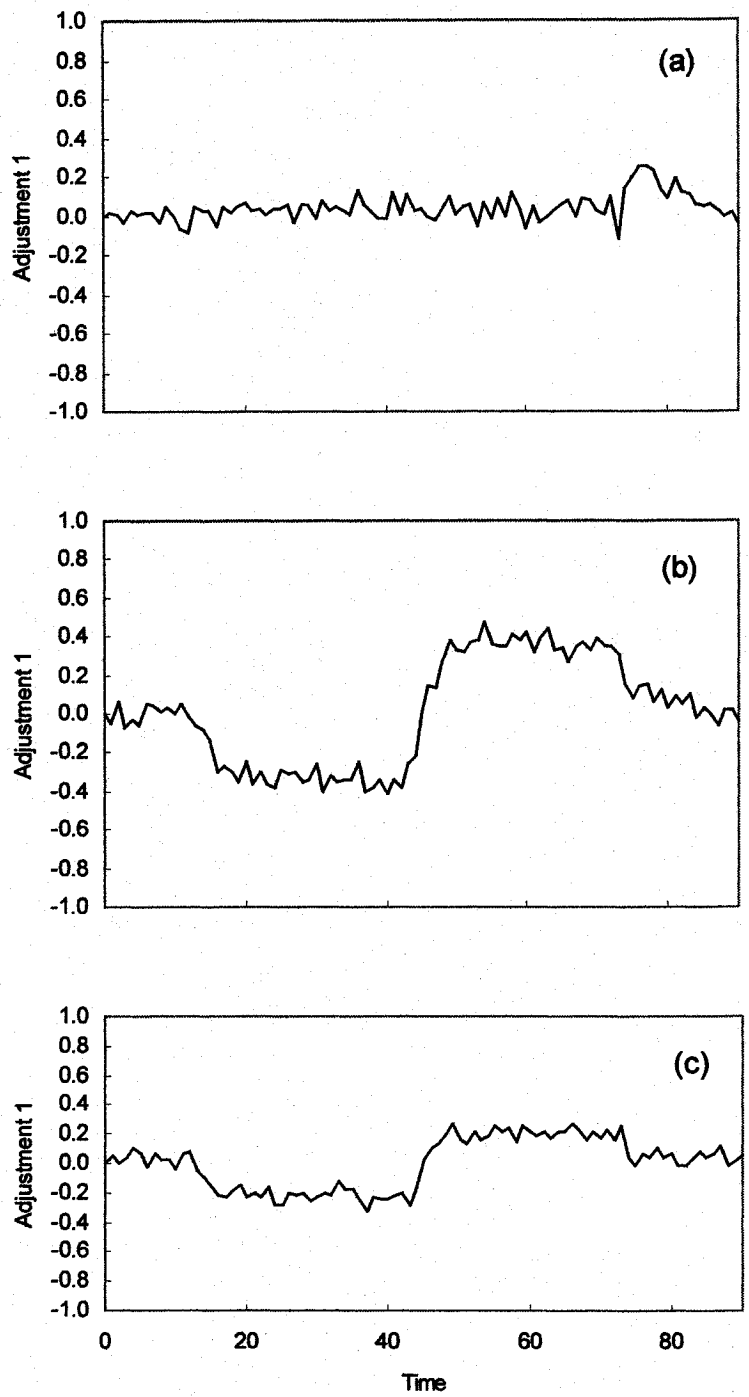

Figure 3-24. Response of the adjustments in y_1 to doublet changes in set-points when $y_1(s)/u_1(s)$ model contains 30% gain mismatch and one of the manipulated variables is

held constant at a time. (a), (b), and (c) show the adjustment in y_1 when u_1, u_2, and u_3, respectively, are held constant for the duration of the doublet.

Case 2: Now let us assume that the $y_2(s)/u_3(s)$ model contains a 30% mismatch in time constant, *i.e.*, $\tau_{model} = 1.3 \times \tau_{process}$. A doublet change in all three set-points is made at time =10. The responses of the normalized adjustment variables are shown in Figure 3-25. Figure 3-25(b) shows that the adjustment for y_2 displays a pattern consistent with time constant mismatch. The adjustment terms for the other two outputs show very minor activity due to the disturbance. Next, the same doublets are applied in the set-points, but u_1, u_2, and u_3, are held constant one at a time over the duration of the set-point changes. The results obtained are shown Figure 3-26. Figure 3-26(c) shows that the pattern previously observed for the adjustment for y_2 is absent when u_3 is held constant. The normalized variance is reduced to 0.0022 from 1.49. When u_1 and u_2 are held constant (each individually) the pattern observed in Figure 3-25(b) reappears, indicating a time constant mismatch in the $y_2(s)/u_3(s)$ model.

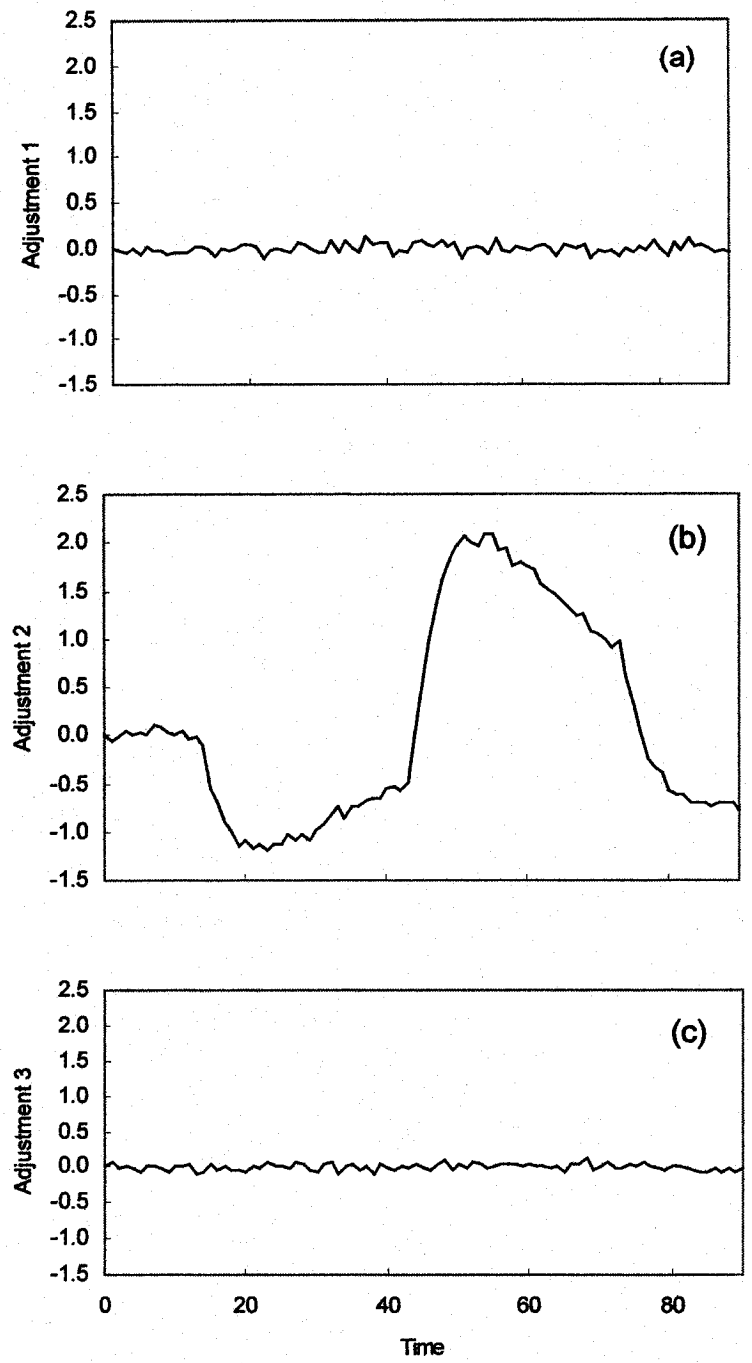

Figure 3-25. Response of the adjustments in the three outputs to doublet changes in set-points when $y_2(s)/u_3(s)$ model contains 30% time constant mismatch.

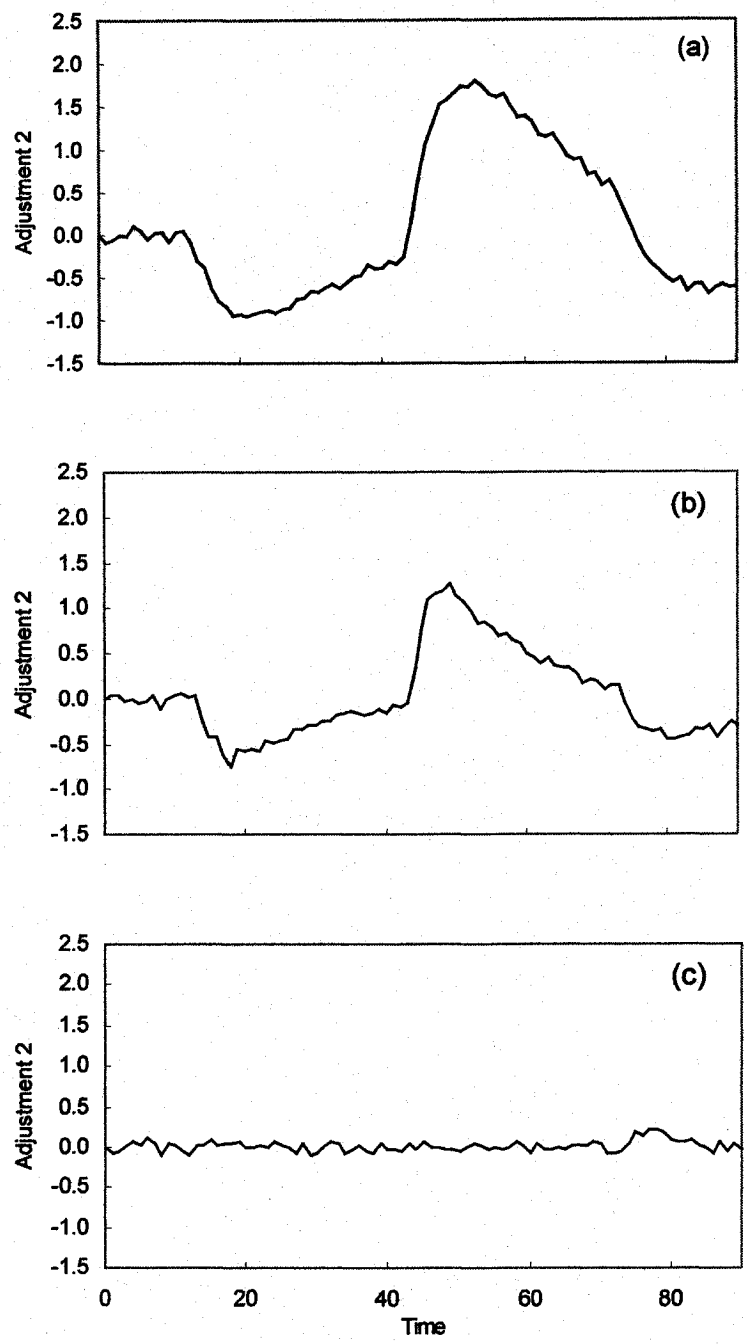

Figure 3-26. Response of the adjustments in y_2 to doublet changes in set-points when $y_2(s)/u_3(s)$ model contains 30% time constant mismatch and one of the manipulated variables is held constant at a time. (a), (b), and (c) show the adjustment in y_2 when u_1, u_2, and u_3, respectively, are held constant for the duration of the doublet.

The previous two examples demonstrated the basic principle of holding the inputs constant to help detect MPM. The method is now adapted to consider a small dithering signal instead of a step change in the set-point, and is combined with the results from Section 3.6.1.

RESULT 3:

Consider an output *i*. If all of the inputs which correspond to a mismatched models for the *i*th row of the trasnsfer function matrix are held, then there will be no correlation between setpoint *r* and ε_i

Proof.

Consider prediction error ε_i, The proof follows directly from :

$$\varepsilon_i = \theta_{i1}(u_1) + \theta_{i2}(u_2) + \theta_{i3}(u_3) + ... + v_i$$

Let J be the index set which contains the elements of mismatched models, i.e $\theta_{ij} \neq 0 \ \ \forall j \in J$. If the u_j are held $\forall j \in J$ then $\varepsilon_i = v_i$ and

$$E(\varepsilon_i r) = E(r v_i) = 0 \qquad \blacktriangle\blacktriangle\blacktriangle$$

Result 4:

If all *u*'s are held except u_i, and only u_{di} is excited, then the correlation between u_{di} and ε_j will be non-zero if and only if $\theta_{ji} \neq 0$.

Proof.

If all u's are held except u_i, and only u_{di} is excited, then

$$\varepsilon_j = \theta_{ji}(u_i + u_{di}) + v_j$$

$$E(\varepsilon_j u_{di}) = E(\theta_{ji} u_i u_{di}) + E(\theta_{ji} u_{di} u_{di}) + E(v_j u_{di}) =$$

Hence,

$$E(\varepsilon_j u_{di}) = 0$$

only if θ_{ji}=0 ▲▲▲

Result 3 may be used to further reduce the subset of candidate models for re-identification. Similar results hold for the correlation between ε_i and u_d if the correspondsing u_d's are set to zero as the manipulated variables are held constant. Result 4 may be used to pinpoint which specific models are mismatched, however there is a significant loss of control involved in holding all except one manipulated variable constant. This one variable will be responsible for controlling all of the ouputs. However, if a diagonal controller is designed, then the free manipulated variable may be used to control one of the outputs considered the most important.

However, as we exapnd on Example 2 from Section 3.6.1, it is not always necessary to hold all possible subsets of the manipulated variables to pinpoint the mismatched models. Consider first holding u_1 constant (from Example 2 it is clear that u_3 is not contributing to any of the mismatch), and set u_{d1} to zero. There is no correlation in Row 3 any more, so this identifies G_{31} as the only mismatched model for y_3. Since we know that G_{23} is not mismatched, then the correlation in Row 2 implies that model G_{22} is mismatched. However we are still uncertain about model G_{21}. To test this model u_2 can

be held. In this case Row 2 and Column 1 showed significant correlation indicating that also model G_{21} is mismatched. Note that this method does not depend on a particular set of parameters being mismatched, but is sensitive to differences between the process and model.

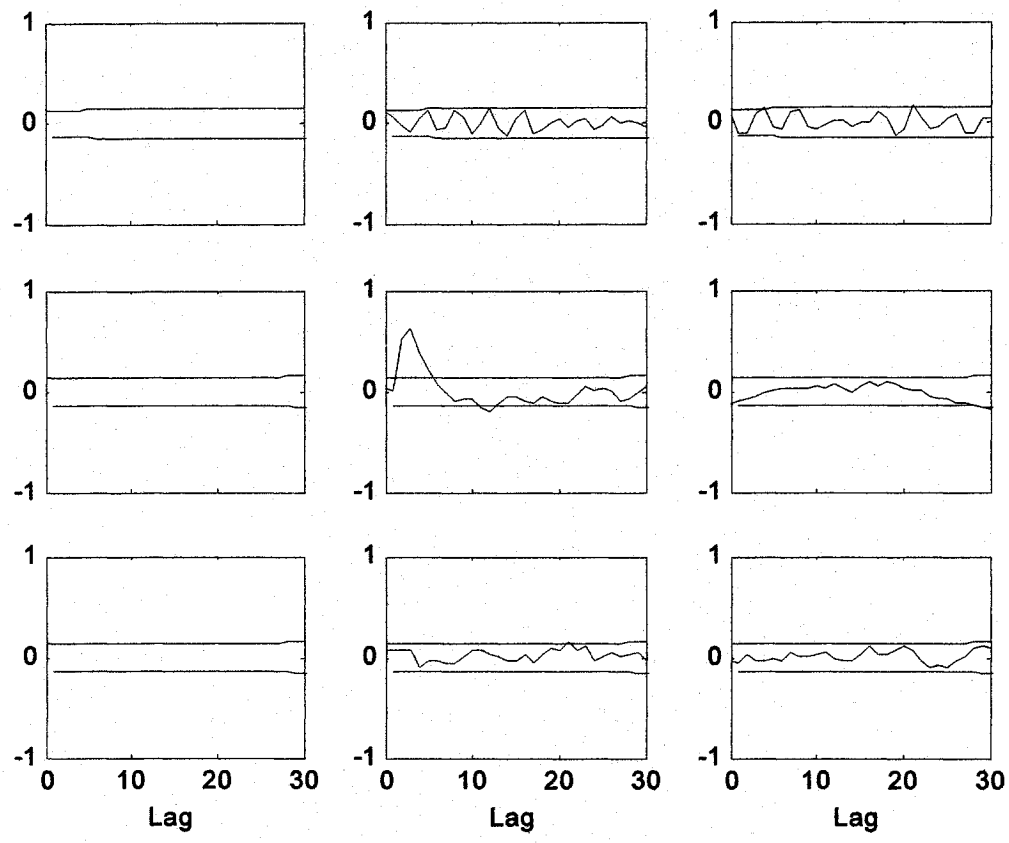

Figure 3-27. Cross-correlations at various lags between ε and dithering signal u_d for example 2, with U1 and ud1 set to zero. The 95% confidence intervals are also shown.

The previous methods provide a yes/no answer concerining mismatch. While yes/no approaches have been the subject of recent articles [12], in practice all models will have some degree of mismatch due to model order reduction or undermodeling. Furthermore, is desirable to have more than a yes or no answer to the degree of mismatch. To answer these questions we can still use the information collected from experiments with external dithering signals in the correlation analysis approach which does not require the control engineer to estimate model orders. This approach is combined with a seive bootstrap to determine confidence bands and is the subject of the next chapter.

CHAPTER 4 DETECTION OF MODEL MISMATCH THROUGH CORRELATION ANALYSIS

Correlation analsysis is a standard non-parametric technique for system identification [16, 89], and has been used in engineering applications [15].It leads to large dimeniosnal models, and is not considered as accurate as prediction error methods [16]. However, a significant advantage is that, other than the truncation point, no model parameterizations need to be made by the practioner. Corrleation analysis was quite popular in the 1960's (e.g [90-92]), but modern focus has been on methods which provide parametric models for controller design [93]. Since DMC and IDCOM directly utilize the step and impuslse response functions, direct estimation of the impulse response function can be useful in these cases. Moreover, for screening model mismatch, a rigourous parametric identification is not needed. A quicker and easier method for screening mismatch is provided by correlation analysis.

One would like to determine if the mismatch is statistically significant. The statistical properties of the 'other' nonparmetric methods (i.e. frequency response methods) in closed-loop identification has been studied in [94, 95]. However, there has been little attention paid to the statistical properties of closed-loop estimates using correlation analysis. Therefore, a bootstrapping method for estimating the uncertainty in the impulse response and process gain is developed in this thesis. The method is first devloped for open-loop SISO systems, then for MIMO open-loop systems, and finally for closed-loop MIMO systems. This method will provide a method for testing which models contain statistically significant errors. Ultimately, this chapter presents a method for estimating the degree of model mismatch in the time domain, and the uncertainty of that estimate.

4.1 SISO open-loop case

Nonparametric finite impulse response (FIR) or step response models, while not parsimonious, are being used in some popular model predictive control schemes such as DMC or IDCOM [76]. FIR models can provide reasonable estimates of the process gain, time delay, and dominant time constant. Further, the estimated impulse response can aid in the selection of suitable parametric models for subsequent fitting via prediction error methods. Impulse or step responses estimated via transient analysis are sensitive to disturbances, and can require inputs with large magnitude. Therefore, impulse and step models are better identified using correlation analysis [15] or least squares optimization [14]. This section considers univariate correlation analysis, and presents a sieve bootstrap method for estimating the uncertainty of the FIR coefficients or functions of the coefficients (e.g. steady state-gain).

It is important to have an estimate for the variance of the coefficients under repeated hypothetical experiments. That is to know how much the coefficients would vary if the experiments were to be repeated under identical conditions,. An analysis for the finite sample impulse response error covariance matrix was developed in [16] for the case of white noise inputs (or prewhitened inputs) . The resulting equation in [16] requires a priori knowledge of the impulse response coefficients. Typically, the estimated impulse response coefficients are substituted for the actual coefficients. The analytical approach and new proposed bootstrap methods are compared via Monte Carlo simulations. If the input is not assumed to be white noise, then an analysis similar to the one in [16] would be considerably more complicated. The bootstrap approach presented here can estimate the variance for either case as it replaces difficult analyses with a computer intensive approach. In the sequel, Monte Carlo simulations are presented which demonstrate that the variance of the non-prewhitening approach can be significantly smaller than the prewhitening approach. And it is found that the bootstrap method can

provide good estimates of the variance. The estimated steady-state gain is usually of importance, but it can be analytically tedious to calculate the variance of the gain from the covariance matrix of the individual impulse response coefficients. Again, this difficulty is easily by-passed via the bootstrap approach.

Bootstrapping was introduced by Efron in 1979 [5] as a flexible method for estimating the sampling distribution of a function of independent observations. Good introductions to bootstrapping methods may be found in [53-56]. One method for handling observations with serially correlated data is to resample blocks of data [96, 97]. However, proper selection of block length is difficult. The other standard method is to reduce the observations to a set of independent and identically distributed (iid) data, and resample from the new iid data (for example see [53, 54, 98, 99]). Since the data have been reduced to independent variates, the random sampling does not destroy any serial correlation. This is typically accomplished by fitting a model whose prediction errors are iid. However, the prediction errors from a model fit via correlation analysis are not iid in the presence of coloured noise. To maintain the non-parametric flavour of correlation analysis, the nonparametric sieve bootstrap [100, 101] is applied to the prediction errors, in which an infinite dimensional autoregressive (AR) process is approximated by a truncated AR model. This allows a computer intensive method to be developed for estimating variance of the impulse response coefficients, or functions thereof such as the process gains.

4.1.1 Correlation analysis

Consider a stable linear time invariant system disturbed by a coloured stochastic process:

$$y(t) = \sum_{i=0}^{N_s} g(i)u(t-i) + v(t) \tag{4.1}$$

where y is the process output, u is the process input, v is the disturbance, and g are the impulse response coefficients to be identified. It is assumed that the disturbance is generated by white noise filtered through a stationary and invertible linear filter.

By multiplying both sides of Eq.(4.1) by lagged inputs (i.e. *u(t-k)*) and taking expectations, it may be shown that [16]:

$$\begin{pmatrix} \hat{c}_{yu}(0) \\ \vdots \\ \hat{c}_{yu}(N_s) \end{pmatrix} = \begin{pmatrix} \hat{c}_u(0) & \cdots & \hat{c}_u(N_s) \\ \hat{c}_u(1) & \ddots & \\ \vdots & & \vdots \\ \hat{c}_u(M) & \cdots & \hat{c}_u(0) \end{pmatrix} \begin{pmatrix} \hat{g}(0) \\ \vdots \\ \hat{g}(N_s) \end{pmatrix} \tag{4.2}$$

where $\hat{g}$ are the estimated impulse response coefficients, and $\hat{c}_{yu}$ and $\hat{c}_u$ are the sample cross covariance and auto covariance, respectively. For a sample size of N, the covariances are estimated via:

$$\hat{c}_{yu}(k) = \frac{1}{N} \sum_{i=1-\min(k,0)}^{N-\max(k,0)} y(t+k)u(t) \tag{4.3}$$

$$\hat{c}_u(k) = \frac{1}{N} \sum_{i=1}^{N-k} u(t+k)u(t)\,. \tag{4.4}$$

If the input is white noise, then Eq. (4.2) simplifies to :

$$\hat{g}(k) = \hat{c}_{yu}(k) / \hat{c}_u(0) \tag{4.5}$$

If the input is not white noise, then the impulse response function can be determined by solving Eq. (4.2), or a pre-whitening approach can be adopted [102]. Let $\psi(z^{-1})$ be some filter that reduces the input to a sequence of iid data, *a(t)*. $\psi(z^{-1})$ is determined using standard time series analysis techniques.

$$\psi(z^{-1})u(t) = a(t) \tag{4.6}$$

The output is filtered using the same filter,

$$\psi(z^{-1})y(t) = b(t) \tag{4.7}$$

The impulse response coefficients are then estimated by:

$$\hat{g}(k) = \hat{c}_{ba}(k) / \hat{c}_a(0) \tag{4.8}$$

The estimator given by Eq. (4.2) and Eq. (4.8) will be referred to as the non-prewhitening and prewhitening approaches, respectively.

For the prewhitening approach an analysis of the finite sample variance was performed in [16]. For a sample size of N, the marginal variance of the k^{th} impulse response coefficient is given as:

$$\text{var}(\hat{g}(k)) \approx \frac{Var(v)}{NVar(u)} + \frac{1}{N}\sum_{\substack{i=0 \\ i\neq k}}^{\infty} g(i)^2 + \frac{1}{N^2}\sum_{\substack{\tau=-k \\ \tau\neq 0}}^{k} (N-|\tau|)g(\tau+k)g(k-\tau) \tag{4.9}$$

Since the actual values of v and g(i) are unknown, they are replaced with their estimated values.

4.1.2 Sieve bootstrap approach

Assume that the impulse response model has been identified. We would now like to estimate the marginal variances of the impulse response coefficients. The main difficulty in applying the bootstrap to the correlation analysis is that the prediction errors are not white noise in the presence of coloured disturbances. Hence, the method of sieve bootstrapping is applied to the estimated disturbance signal. The disturbance, *v(t),* may be estimated by:

$$\hat{v}(t) = y(t) - \sum_{i=0}^{Ns} \hat{g}(i)u(t-i) \tag{4.10}$$

Next the disturbance is modeled by using the method of sieves. The consideration of semi-nonparametric models, such as infinite AR or MA models, leads to infinite dimensional parameter spaces. For tractability, the infinite dimensional parameter space is approximated by a series of finite dimensional spaces referred to as sieves. The optimal approximation is determined by some criteria function. Consider a system that can be described by an infinite dimensional AR model. Using the Akaike information criteria (AIC) an optimal finite dimensional AR model is selected from a sequence of different order AR models. Thus, a stable AR time series model, $\zeta(z^{-1})$, is fit to $\hat{v}(t)$.

$$\zeta(z^{-1})\hat{v}(t) = \hat{a}(t) \tag{4.11}$$

where $\hat{a}(t)$ is a sequence of iid data. Since the $\hat{a}(t)$ will be approximately independent, they may be randomly sampled without destroying any serial correlation. Before the data are collected for correlation analysis it is important to allow the system to achieve stationarity. This is done via a "burn-in" process in which the system is excited for a certain number of sampling instants (N_b) before the data are collected. The following procedure may now be used to estimate the variance for a sample size N:

1) Set $j = 1$. Generate a realization of the input, $u(i)_j^*$, of length (N_b+N).

2) Randomly sample with replacement (N_b+N) samples from $\hat{a}(t)$. Call the randomly selected set $\hat{a}(t)_j^*$. Next a pseudo realization of $v(t)$ is generated by:

$$\hat{v}(t)_j^* = \zeta^{-1}(z^{-1})\hat{a}_j^*(t) \tag{4.12}$$

3) Create a pseudo realization of the process output via:

$$y(t)_j^* = \sum_{i=0}^{N_s} \hat{g}(i) u_j^*(t-i) + v_j^*(t) \tag{4.13}$$

4) The correlation analysis is then applied to the last N values of $u(i)_j^*$ and $y(i)_j^*$, which results in the j^{th} bootstrap set of estimated impulse response coefficients, $\hat{g}(i)_j^*$.

5) Steps 1) to 4) are repeated B times. The marginal variances are then estimated as:

$$\text{var}(\hat{g}(k))_{BS} = \frac{1}{B-1}\sum_{j=1}^{B}(\hat{g}(k)_j^* - \frac{1}{B}\sum_{i=1}^{B}\hat{g}(k)_i^*)^2 \tag{4.14}$$

If desired, the complete error covariance of the impulse response coefficients can also be determined at step 5, but this thesis considers only the marginal variances. Note further, that no parametric assumptions about the disturbance are required to use this method.

Frequently the steady state gain is of interest, hence it is important to assess the uncertainty of the estimated gain. The standard error of the estimated gain, $\hat{k}$, can be found from:

$$se(\hat{k})_{BS} = \sqrt{\frac{1}{B-1}\sum_{j=1}^{B}(\hat{k}_j^* - \frac{1}{B}\sum_{i=1}^{B}\hat{k}_i^*)^2} \tag{4.15}$$

where,

$$\hat{k}_j^* = \sum_{i=1}^{M}\hat{g}(i)_j^* \tag{4.16}$$

4.1.3 Examples

This section presents several computer-simulated examples based on the following system:

$$y(z) = \frac{0.09516}{z^5 - 0.9048\, z^4} z^{-4} u(z) + v(z) \tag{4.17}$$

$$v(z) = \frac{1-.4z^{-1}}{1-.6z^{-1}} a(z). \tag{4.18}$$

Where a is normally distributed with mean zero, and a variance of 0.1. Several different standard inputs will be considered, *viz* white noise, autoregressive moving average filtered white noise (ARMA), and pseudo-random binary sequences (PRBS). In all cases the settling time was taken as $N_s = 80$. When fitting the AR model to the disturbance, no model order greater than two was used.

First consider the simple case of an input, u, randomly drawn from a $N(0,1)$ distribution (normally distributed with mean zero and variance one). For a sample size of $N = 1000$, Monte Carlo simulations were run with 10,000 repetitions. The Monte Carlo simulations provide the 'gold-standard' for comparison. The resulting marginal variances of the FIR coefficients are plotted as a function of lag in Figure 4-1. In this case both Eq. 9 and the sieve bootstrap provide good estimates of the marginal variances. However, when the sample size is reduced to $N = 200$, Eq. (4.9) over predicts the variance for larger lags, while the sieve bootstrap captures the trend in the variance (Figure 4-2).

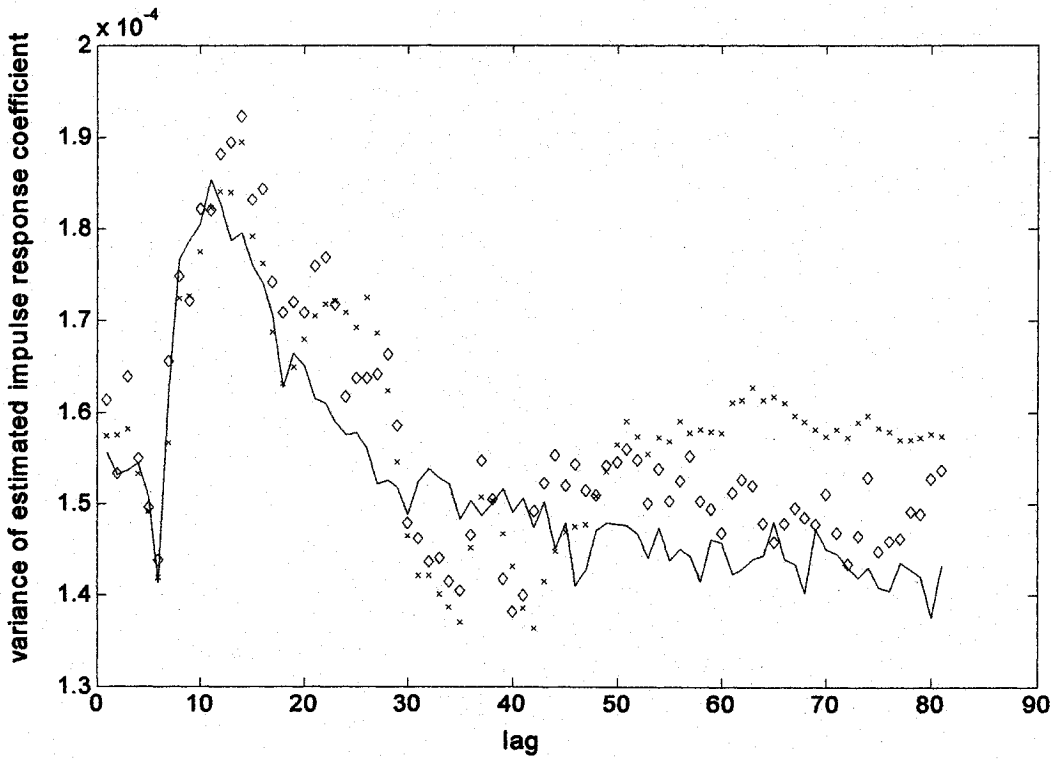

Figure 4-1. Variance for white noise input, $N = 1000$. — : Monte Carlo, x: Eq. **(4.9)**, ◊: sieve bootstrap.

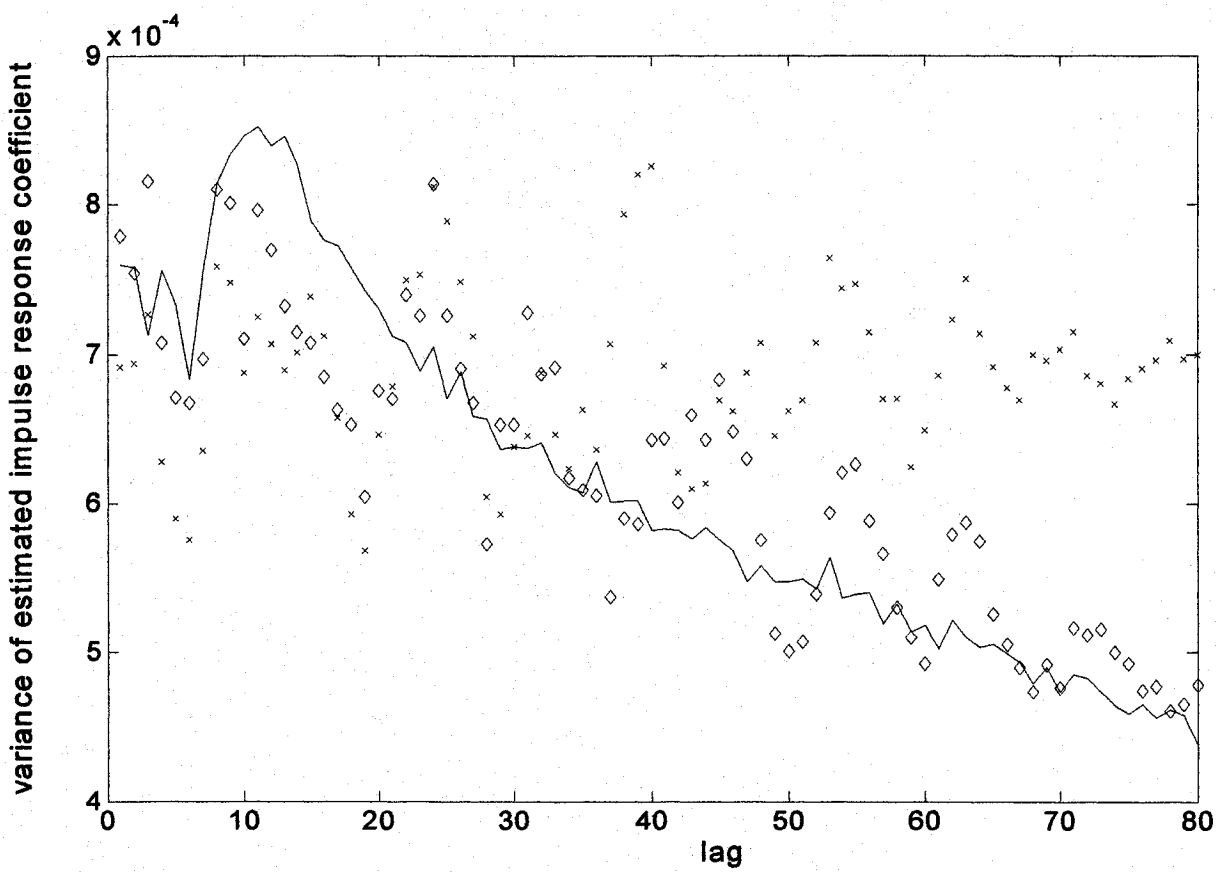

Figure 4-2. Variance for white noise input, $N = 200$. — : Monte Carlo, x: Eq. **(4.9)**, ◊: sieve bootstrap.

While the simplified form Eq. (4.8) is easy to apply, the full form, Eq. (4.2), is demonstrated to provide a variance in the following examples. Consider an input randomly sampled from a $N(0,1)$ distribution. For a sample size of $N = 500$, the results are shown in Figure 4-4. The use of Eq. (4.2) resulted in up to a 40% reduction in the variance. For both cases the bootstrap method again accurately estimates the variance. Next consider a random input described by the following time series ARMA model:

$$u(z) = \frac{1-.8z^{-1}}{1-.5z^{1}} a(z) \tag{4.19}$$

In this case Eq. (4.8) is applied to the input and output pre-filtered by the inverse of the system described by Eq.(4.19). Solving the full set of linear equations (Eq. (4.8)) significantly reduced the variance at the extreme lags (Figure 4-5). Having a lower variance at the initial lags would be helpful when estimating the process delay. Also, note that Eq. (4.9) did not provide less accurate estimates of the variance.

Since the estimated FIR coefficients are not independent, it can be tedious to calculate the variance of the steady state gain from the covariance matrix of the FIR coefficients. Table 4-1 contains the true, and bootstrap estimates of the variance for the steady state gain when using Eq. (4.8) to solve for the FIR coefficients. Note that for larger samples the bootstrap provides good estimates. However for the sample size of $N = 200$ the estimate was not reliable. For a system with a settling time of 80, the sample sizes of $N = 500$ or 1000 would be more appropriate.

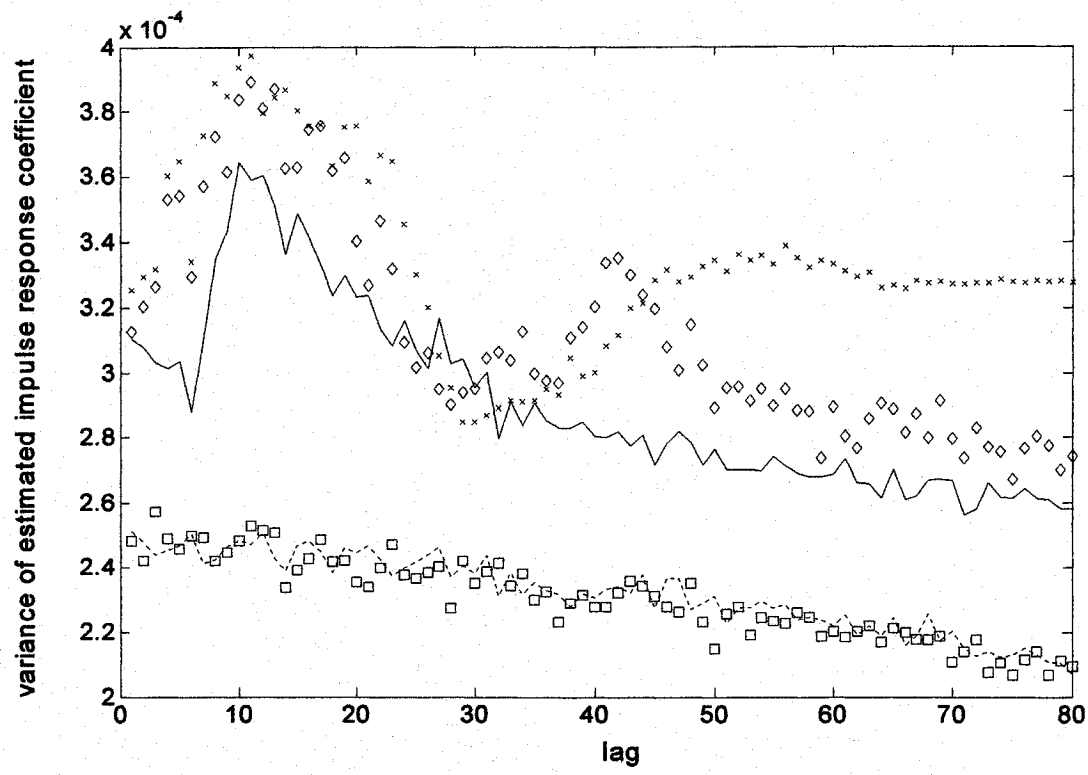

Figure 4-4. Variance for white noise input, $N = 500$. — : Monte Carlo (Eq. (4.8)), x: Eq. (4.9), ◊: sieve bootstrap (Eq. (4.8)), ---: Monte Carlo (Eq.(4.2)), □: sieve bootstrap (Eq. (4.2)).

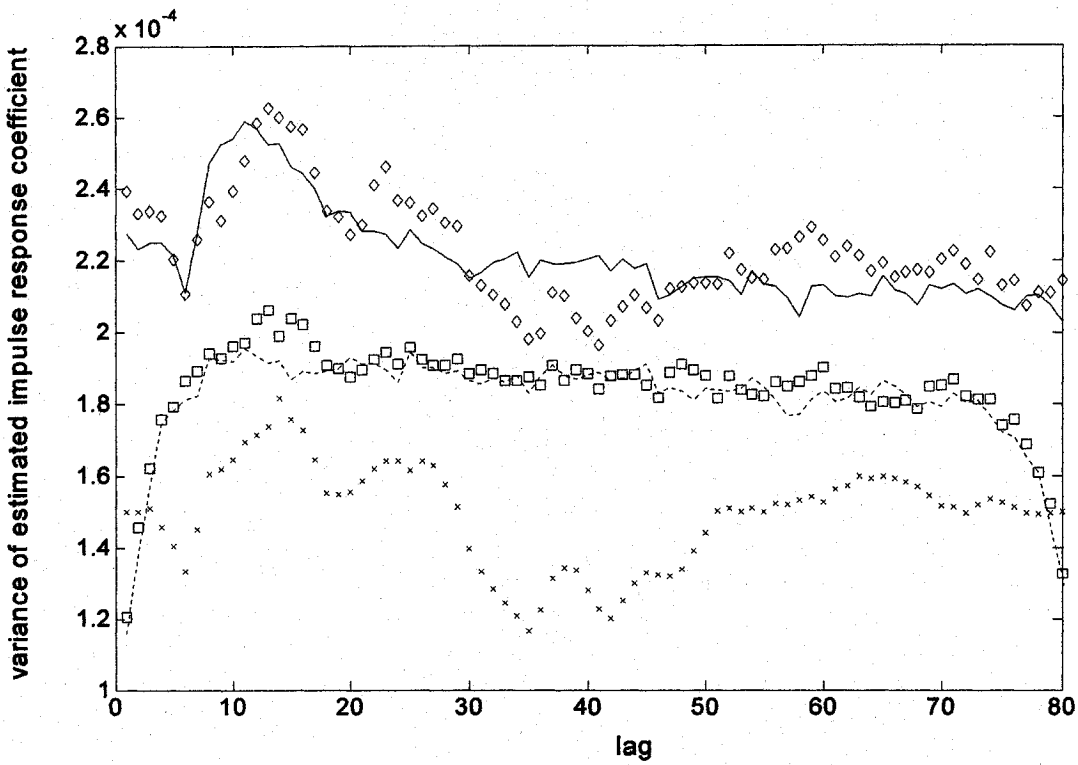

Figure 4-5. Variance for ARMA, $N = 1000$. — : Monte Carlo (Eq. (4.8)), x: Eq. (4.9), ◊: sieve bootstrap (Eq. (4.8)), ---: Monte Carlo (Eq. (4.2)), □: sieve bootstrap (Eq. (4.2)).

Table 4-1. Estimated and true standard deviation for steady state gain

	$N = 200$	$N = 500$	$N = 1000$
$se(\hat{k})$	0.45	0.41	0.30
$se(\hat{k})_{BS}$	0.18	0.45	0.26

4.2 MIMO open-loop case

Traditionally, multivariate correlation analysis is applied in a univariate fashion by considering one input at a time [14]. However, to minimize the amount of time required to perform the system identification experiments, and to account for possible input correlation, it is desirable to apply the correlation analysis in a fashion that explicitly accounts for the multivariate nature of the problem. Hence, the multivariate form of correlation analysis has been derived in [103, 104].

Just as in the univariate case, prewhitening of the inputs can lead to a simplification of the correlation analysis. It has been observed that using random binary inputs provides some improvement over the non-prewhitening approach [104]. However the statistical performance of multivariate correlation analysis approaches was not considered in [104]. This section extends the previous sieve bootstrap method to estimate the uncertainty bands of multivariate impulse response functions and the confidence limits for the process gain matrix determined via correlation analysis.

The multivariate case can be treated as a number of separate multiple-input/single-output identifications. It can be shown that [104]:

$$C_u \bullet g = C_{yu} \tag{4.20}$$

where C_u is a auto-covariance matrix of the inputs, g is a vector of impulse response coefficients to be estimated, and C_{yu} is a cross-covariance matrix of the inputs and the output under consideration. If we have N_u inputs, then the above variables are described as:

$$h^T = [h_{01}\ h_{11}\ h_{21} \cdots h_{m1}\ h_{02}\ h_{12}\ h_{22} \cdots h_{mp}\ h_{0p}\ h_{1p}\ h_{2p} \cdots h_{mp}] \tag{4.21}$$

$$C_{yu}{}^T = [C_1\ C_2\ C_3 \ldots C_p] \tag{4.22}$$

$$C_j{}^T = [c_{x_j y}(0)\ c_{x_j y}(1) \ldots . c_{x_j y}(m)] \tag{4.23}$$

$$C_u{}^T = [C_{u1}\ C_{u2}\ C_{u3} \ldots C_{uN_u}] \quad C_{ui} = [C_{ui1}\ C_{ui2}\ C_{ui3} \ldots C_{uip}] \tag{4.24}$$

$$C_{uij} = \begin{bmatrix} c_{x_i y_j}(0) & c_{x_i y_j}(-1) & c_{x_i y_j}(-2) & \cdots & c_{x_i y_j}(-N_s) \\ c_{x_i y_j}(1) & c_{x_i y_j}(0) & c_{x_i y_j}(-1) & \cdots & c_{x_i y_j}(-N_s+1) \\ c_{x_i y_j}(2) & c_{x_i y_j}(1) & c_{x_i y_j}(0) & \cdots & c_{x_i y_j}(-N_s+2) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ c_{x_i y_j}(N_s) & c_{x_i y_j}(N_s-1) & c_{x_i y_j}(N_s-2) & \cdots & c_{x_i y_j}(0) \end{bmatrix} \tag{4.25}$$

Thus, once the co-variances from Equation (4.25) and the auto-covariances from Equation (4.23) are estimated (via their sample counterparts) the impulse response coefficients may be calculated by solving Equation (4.20). Note that g_{ij} represents the *i*th impulse response coefficient between the output under consideration and the *j*th input.

The sieve bootstrap method will essentially be the same as outlined in Section 4.1.2; however the AR model fit to the disturbance estimates will become a vector autoregressive model (VAR). Also special attention must be accorded to the sampling of the residuals. VAR models are commonly used to model stationary multivariable time series. They are flexible models parameterized by a single lag parameter *b*. Let $\Upsilon(t)$ represent a N_Υ dimensional vector of multivariable time series observations:

$$\Upsilon(t) = \begin{pmatrix} \Upsilon_1(t) \\ \Upsilon_1(t) \\ \vdots \\ \Upsilon_{N_\Upsilon}(t) \end{pmatrix} = \mathrm{A}_1\Upsilon(t-1) + \mathrm{A}_2\Upsilon(t-2) + \ldots + \mathrm{A}_p\Upsilon(t-b) + a(t) \tag{4.26}$$

where A_1 is a $N_\Upsilon \times N_\Upsilon$ coefficient matrix, and $a(t)$ is a $N_\Upsilon \times 1$ column vector of mean zero white noise with covariance matrix Σ. For example a VAR with three observed time series with a lag of b =2 would be given as:

$$\begin{pmatrix} \Upsilon_1(t) \\ \Upsilon_2(t) \\ \Upsilon_3(t) \end{pmatrix} = \begin{pmatrix} \alpha_{11}^1 & \alpha_{12}^1 & \alpha_{13}^1 \\ \alpha_{21}^1 & \alpha_{22}^1 & \alpha_{23}^1 \\ \alpha_{31}^1 & \alpha_{32}^1 & \alpha_{33}^1 \end{pmatrix} \Upsilon(t-1) + \begin{pmatrix} \alpha_{11}^2 & \alpha_{12}^2 & \alpha_{13}^2 \\ \alpha_{21}^2 & \alpha_{22}^2 & \alpha_{23}^2 \\ \alpha_{31}^2 & \alpha_{32}^2 & \alpha_{33}^2 \end{pmatrix} \Upsilon(t-2) + \begin{pmatrix} a_1(t) \\ a_2(t) \\ a_3(t) \end{pmatrix} \tag{4.27}$$

where

$$\mathrm{A}_i = \begin{pmatrix} \alpha_{11}^i & \alpha_{12}^i & \alpha_{13}^i \\ \alpha_{21}^i & \alpha_{22}^i & \alpha_{23}^i \\ \alpha_{31}^i & \alpha_{32}^i & \alpha_{33}^i \end{pmatrix}$$

The VAR coefficients are usually determined via ordinary least squares. The number of lags, b, to include can be determined by the AIC and verification that each of the estimated residuals are independent (*i.e.*, white noise). Further details on VAR modeling may be found in [105].

Care must be taken when performing the bootstrap resampling of the residuals. Each individual residual should be independent (e.g., a_1 is independent white noise), however this does not mean that the various residuals do not have some cross-correlation (i.e., a_1 and a_2 may not be independent). To account for this cross-correlation the following sampling procedure is used for a sample size of N:

1) Let j be a N dimensional vector of random numbers between 1 to N.

2) The bootstrap residuals should then be $a_{bootstrap} = [\hat{a}_1(j)\ \hat{a}_2(j)\ \hat{a}_3(j)]$ where j is used as an indexing vector.

For example, consider a sample size of 5 points. The estimated residuals turn out to be:

$$\hat{a}=\begin{pmatrix} 4.3 & .4 & 1 \\ 2.2 & -.1 & -1.2 \\ -3.2 & .2 & 2.1 \\ -1.3 & .05 & .5 \\ 5 & .12 & -1 \end{pmatrix} \tag{4.28}$$

A 5-dimensional vector of random numbers from 1 to 5 is generated, say:

$$j=\begin{pmatrix} 3 & 1 & 4 & 3 & 2 \end{pmatrix}^T \tag{4.29}$$

Therefore the bootstrap samples of the residuals will be:

$$a_{bootstrap}=\begin{pmatrix} -3.2 & .2 & 2.1 \\ 4.3 & .4 & 1 \\ -1.3 & .05 & .5 \\ -3.2 & .2 & 2.1 \\ 2.2 & -.1 & -1.2 \end{pmatrix} \tag{4.30}$$

Example 1:

Consider the Shell oil fractionator 3×3 system adapted from [76]:

$$G=\begin{pmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.72e^{-14s}}{60s+1} & \frac{6.90e^{-15s}}{40s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.2}{19s+1} \end{pmatrix} \tag{4.31}$$

A stochastic disturbance is modeled as:

$$v=Ha=diag(\frac{1.2}{45s+1},\frac{1.52}{25s+1},\frac{1.14}{27s+1})a \tag{4.32}$$

where a is a 3 dimensional vector of mean zero white noise with covariance matrix of diag(0.5,0.5,0.5). The inputs are excited by a mean zero white noise input with covariance matrix diag(1,1,1). Both a and the input u are normally distributed. The results for a sample size of $N = 3000$ are shown in Figure 4-6. 10,000 repetitions were used in the Monte Carlo simulations to determine the true distribution, and 4000 bootstrap replications were used. Note that the bootstrap provided a very close approximation to the true uncertainty bound. Table 4-2 shows the bootstrap estimates of the true standard deviation of the process gains compared to the true standard deviations (determinded via Monte Carlo). Reasonable estimates of the standard deviation are provided by the bootstrap method.

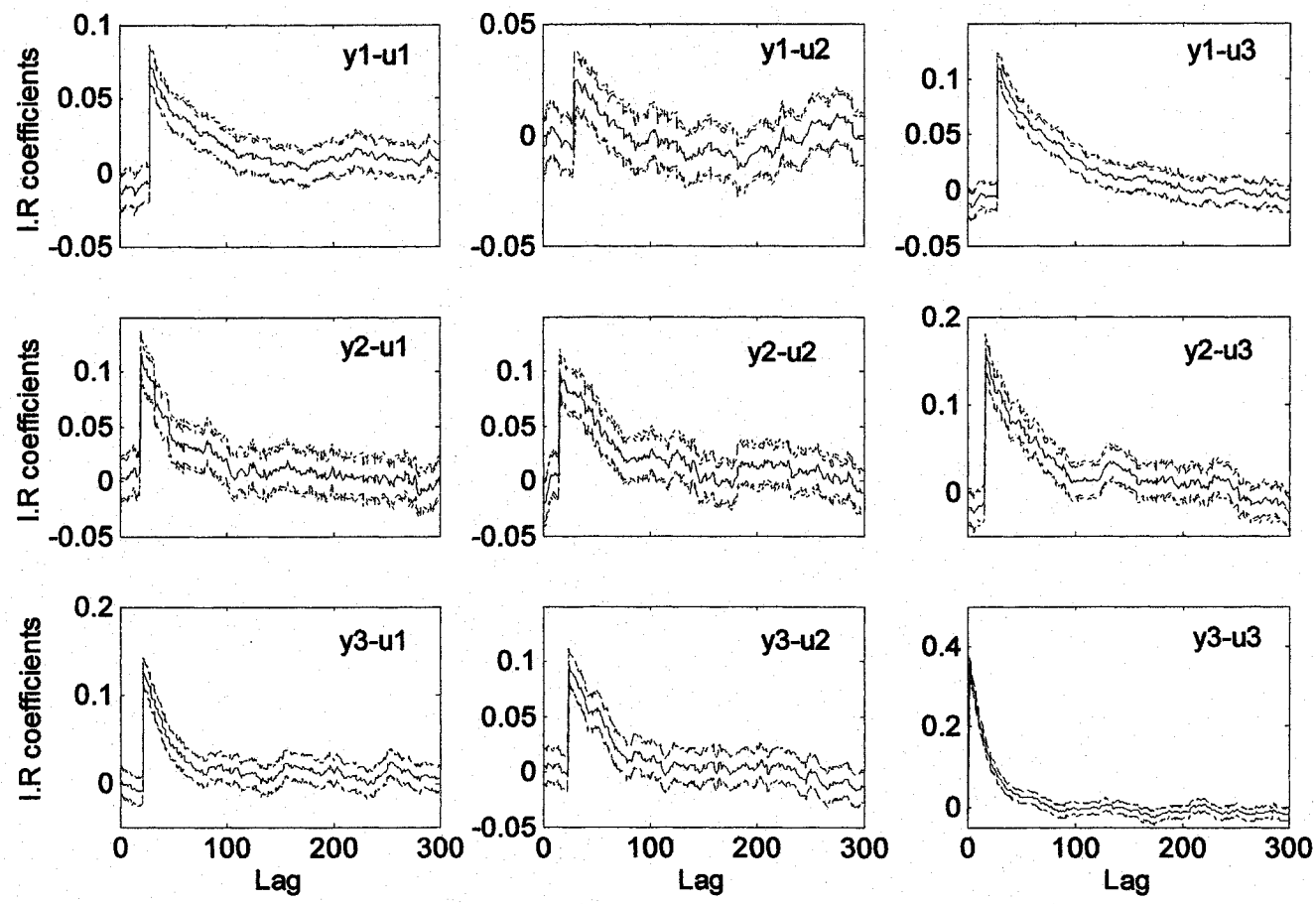

Figure 4-6. Estimated impulse response and their associated 95% confidence bounds determined via Monte Carlo (dashed) and bootstrap simulation (dash-dot).

Table 4-2. Estimated and true standard deviation for steady state gain

	y_1	y_2	y_3
u_1	0.97, 0.82	0.96, 0.80	0.98, 0.82
u_2	1.27, 0.96	1.27, 0.99	1.28, 0.99
u_3	0.94, 1.04	0.95, 1.00	0.96, 1.00

$se(\hat{k})_{BS}$, $se(\hat{k})$

In Figure 4-7 one can see how the confidence bands can help identify the process delay. The estimated impulse response coefficient will be noisy and it will be difficult to tell where the dead time occurs, but with the uncertainty bands the first coefficnet which is statistically significant indicates the delay. Also, since one has the many different realizations of the impulse response coefficents from the bootstrap simulations, the uncertainty in the frequency domain may be obtained, which can be useful for robust controller design.

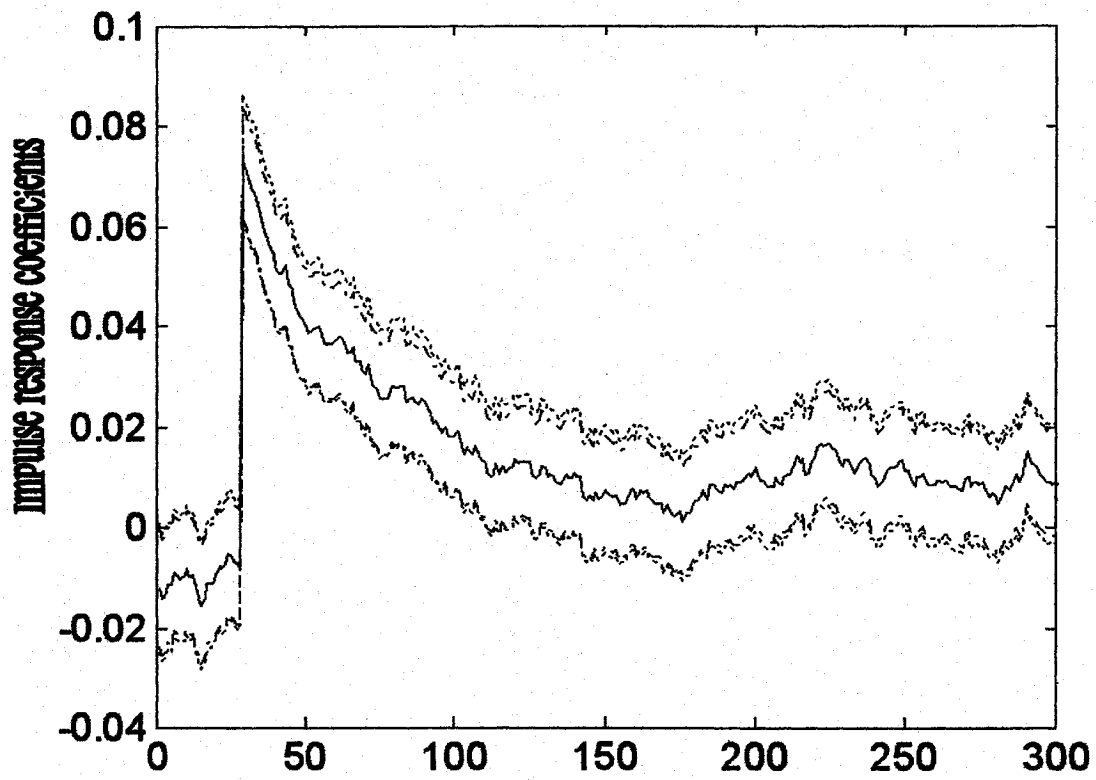

Figure 4-7. Estimated impulse response (solid) and the true 95% confidence bounds determined via Monte Carlo (dashed) and estimated bootstrap confidence interval (dash-dot) for the y_1-u_1 model.

4.3 MIMO closed-loop identification of model mismatch

Closed loop identification via correlation analysis has received little attention in the literature. The statistical properties of the 'other' nonparmetric methods (i.e. frequency response methods) in closed-loop identification has been studied in [94, 95]. However, there has been little attention to the statistical properties of closed-loop estimates using correlation analysis. Thus this section develops a seive bootstrap method suitable for estimating the uncertianty in closed-loop identification when using correlation analysis. The standard methods of closed loop identification include the direct, indirect, and joint-input-output methods [14, 16]. The direct approach essentially ignores the presence of feedback and directly fits a model between the input and process outputs. It is well known that if the direct method is applied to correlation analysis that the inverse of the controller is identified, thus the process is unidentifiable [106]. The indirect method utilizes information about the controller to transform an indentified closed-loop transfer function into the process transfer function. The disadvantage of this method is that the controller transfer function must be known. Since this may not be true for mutlivariable MPC we shall use the joint input-output method. In the joint input-output method two closed-loop transfer functions are identified from a single experiment, and they are combined to provide an estimate for the process model. The basic identification method proposed in this section is described below:

An estimate for the misamtch matrix, $\hat{\theta}$, may be obtained by the following steps:

1) Identify transfer function matrix between the setpoint r and manipilated variable u

$$u = \hat{\Omega} r \tag{4.33}$$

2) Identify a transfer function between setpoint and prediction error.

$$\varepsilon = \hat{\Pi}\, r \tag{4.34}$$

3) Calculate the inverse of $\hat{\Omega}$

$$\hat{\theta} = \hat{\Pi}\hat{\Omega}^{-1} \tag{4.35}$$

Proof.

$$u = (I + Q\theta)^{-1} Q\, r \approx \Omega r$$

$$\Omega^{-1} \approx \left(Q(I + \theta Q)^{-1}\right)^{-1} = (I + \theta Q)Q^{-1}$$

Now,

$$\varepsilon = (I + \theta Q)^{-1}\theta Q = \theta Q(I + \theta Q)^{-1} \approx \hat{\Pi}\, r$$

Therefore,

$$\hat{\Pi}\hat{\Omega}^{-1} \approx \theta Q(I + \theta Q)^{-1}(I + \theta Q)Q^{-1} = \hat{\theta}$$

Only one set of experimetns is required by exciting all three setpoints at once and performing a multivariable correlation analysis. This feature allows time to be saved.

The resulting uncertainty will be estimated using the seive bootstrap approach. The seive bootstrap method for the closed-loop identification is as follows:

1) Excite the setpoints, and then using multivariable correlation analaysis fit models $\hat{\Omega}$ and $\hat{\Pi}$. Refer to Equations (4.34) and (4.35).

2) Estimate the disturbance variates via:

$$v^1(t) = \varepsilon(t) - \hat{\Pi}\, r(t) \tag{4.36}$$

$$v^2(t) = u(t) - \hat{\Omega}\, r(t) \tag{4.37}$$

where the above have approriate matrix dimensions for the system under consideration.

3) Using the AIC fit two AR time series models to the disturbances.

$$\mathrm{A}^1(q^{-1})v^1(t) = a^1(t) \tag{4.38}$$

$$\mathrm{A}^2(q^{-1})v^2(t) = a^2(t) \tag{4.39}$$

where A^1 and A^2 are the AR time series model coefficent matrices. a^1 and a^2 are the time series residuals. The residuals are calculated and stored for later resampling.

4) Bootstrap realizations of the closed loop prediction error, $\varepsilon(t)$, and input, $u(t)$,are found by :

$$\varepsilon(t)_{bootstrap} = \hat{\Pi} r(t) + v^1{}_{bootstrap}(t) \tag{4.40}$$

$$u(t)_{bootstrap} = \hat{\Omega} r(t) + v^2{}_{bootstrap}(t) \tag{4.41}$$

where $v^i{}_{bootstrap}$, i =1 or 2, are the boostrap realizations of the disturbances. They are generated as follows:

Use the resampling scheme described in Section 4.2 to generate boostrap sampling of residuals $a^1(t)$. Use the same random indexing vector to select samples from $a^2(t)$. Call the bootstrap residuals $a^1{}_{bootstrap}(t)$ and $a^2{}_{bootstrap}(t)$.Finally generate the disturbance variates :

$$v^1{}_{bootstrap}(t) = \left(\mathrm{A}^1(q^{-1})\right)^{-1} a^1{}_{bootstrap}(t) \tag{4.42}$$

$$v^2{}_{bootstrap}(t) = \left(\mathrm{A}^2(q^{-1})\right)^{-1} a^2{}_{bootstrap}(t) \tag{4.43}$$

5) Finally using mutlivariable correlation analysis, fit models $\hat{\Omega}_{bootstrap}$ and $\hat{\Pi}_{bootstrap}$ to Equations (4.34) and (4.35). Then generate an estmate of the model mismatch via:

$$\hat{\theta}_{bootstrap} = \hat{\Pi}_{bootstrap}\hat{\Omega}^{-1}_{bootstrap} \tag{4.44}$$

Steps 4 and 5 are repeated as many times as desired. This produces a set of realizations of $\hat{\theta}_{bootstrap}$ which mimics the real world variation in $\hat{\theta}$. Then confidence intervals and variances may be calculated using the set of realizations of $\hat{\theta}_{bootstrap}$.

Example 3.

Consider again the Shell oil fractionator 3×3 system adapted from [76]:

$$G=\begin{pmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.72e^{-14s}}{60s+1} & \frac{6.90e^{-15s}}{40s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.2}{19s+1} \end{pmatrix} \tag{4.45}$$

A stochastic disturbance is modeled as:

$$v=Ha=diag(\frac{1.2}{45s+1},\frac{1.52}{25s+1}\frac{1.14}{27s+1})a \tag{4.46}$$

where a is a 3×1 column vector of mean zero white noise, and covariance matrix $\Sigma_a = diag(0.1,0.1,0.1)$, H is the disturbance transfer function. A zero-order hold with a sampling time of T = 10 is used to discretize the system. An unconstrained DMC controller with a prediction horizon of 50 and control horizon of 3 intervals is used to control the system. The diagonal elements of the dynamic matrix are multiplied by a factor of 1.01 for move suppression. However the models used to design the DMC controller contained mismatch in the following models:

$$\hat{G}_{21}=0.8*\frac{5.39e^{-18s}}{50s+1},\ \hat{G}_{31}=\frac{4.38e^{-18s}}{33s+1},\ \hat{G}_{32}=1.3\frac{4.42e^{-22s}}{44s+1}$$

Therefore the y_2-u_1 model has 20% gain mismatch, the y_3-u_1 model has dead-time mismatch, and the y_3-u_2 model has 30% gain mismatch. A dithering signal on the setpoints of mean zero normally distributed variates with co-variance of diag(0.5,0.5,0.5) was applied simultaneously, and 3000 data points of closed loop data were collected.

Impulse response models were fit between the dithering signal and the prediction error, and also between the dithering signal and the manipulated variables u. The resulting model mismatch $\hat{\theta}$ and the true 95% confidence band (determined via 6000 Monte Carlo simulations) and the bootstrap estimates using 4000 bootstrap replications is shown in Figure 4-8. The bootstrap intervals were determined using the normal-bootstrap method described in Section 2.2.3. Note that the uncertainty estimated by the bootstrap is very close to the true uncertainty band. The models with no mismatch can be identified by examining the responses which are not statistically different from zero (i.e. their uncertainty bands always contain zero). The models with mismatch are clearly identified in Figure 4-8. Since the confidence bands are also presented, the engineer may decide to use the estimated model error to correct the appropriate models if they decide that the uncertainty bands are small enough to have strong confidence in the results. If large confidence bands are present, then the results can be used to indicate which models require re-identification via a more accurate method. This is an advantage over the previous methods since the model error is quantitatively determined along with the uncertainty of the estimates. Note, that after the responses had settled down, it was observed that the impulse response coefficients would sometimes diverge towards infinity. Since this happened after the responses had already settled it is of little consequence. This may be a result of imperfect unstable zero/pole cancellation between the two estimated closed-loop transfer functions.

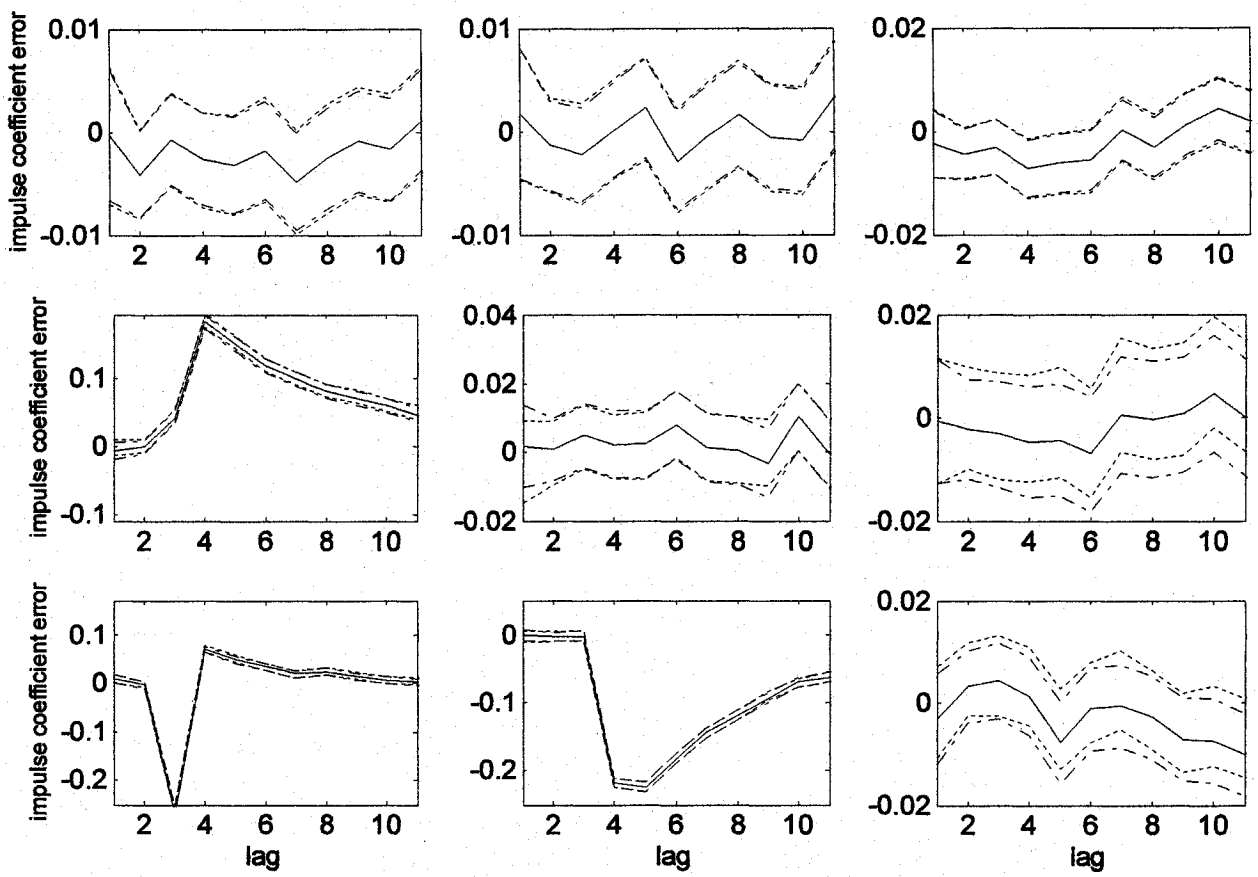

Figure 4-8. Estimated error of impulse response coefficient, and the true (solid) and bootstrap estimates (dotted)

CHAPTER 5 CONCLUSIONS AND FUTURE WORK

The statistical properties of the minimum variance controller index have been estimated using bootstrapping techniques. These estimates need to be made by using small samples since in practice the process data may only be stationary over short intervals. The proposed procedures have demonstrated the ability to provide reasonable estimates by using small sample sizes in case of normal and non-normal innovations. These methods have broader applicability as demonstrated by estimating the variance of the closed-loop settling time performance index while accounting for model order uncertainty. The bootstrap estimated sampling distributions provided reasonable approximations of the experimentally determined distribution functions.

The detection and correction of model plant mismatch has also been addressed in this thesis. The iterative methods for univariate model based controllers can detect and correct gain and dead-time mismatch. For multivariate cases, the cross-correlations and partial control methods are used to detect the input-output pairs that are mismatched . The proposed sieve bootstrap method can estimate the confidence bands for the estimated impulse response functions in both open and closed-loop situations. The broader applicability of the proposed method also allows for estimates of the process gains to be determined. The confidence bands can be used to detect the presence and extent of statistically significant model errors.

Future work on bootstrapping the controller performance indices could include methods to reduce the computational effort, so that the procedures may be implemented in real time. In cases where the controller tuning is identified as the cause of poor control performance, it would also be interesting to use the performance index in some fashion to suggest which controller parameters should be retuned. For example, a method could be developed that suggests that the controller gain should be increased or decreased based on observation of the performance index. It would also be useful to be able to make the

decision to re-identify a process model (and redesign the model based controller) as an optimization problem. The optimization problem would balance the expected increase in controller performance against the time and effort of the re-identification process. Also, it could consider the uncertainty inherent in the identification method. For example, if it is determined that the model gain is three percent too high, but the identification method can only estimate the gain within 10%, then it is not reasonable to re-identify the process model. Bootstrapping could play an important role in this type of decision-making.

REFERENCES

[1] Harris T. J., Assessment of control loop performance, Can. J. Chem. Eng. 67 (1989) 856-861.

[2] Harris T. J., Statistical properties of quadratic-type performance indices, Journal of Process Control. 14 (2004) 899-914.

[3] Horch A. and Isaksson, A. J., A modified index for control performance assessment, Journal of Process Control. 9 (1999) 475-483.

[4] Desborough L. and Harris, T. J., Performance assessment measures for univariate feedback control, Can. J. Chem. Eng. 70 (1992) 1186-1197.

[5] Efron B., Bootstrap methods: Another look at the jacknife, Ann. Statist. 7 (1979) 1-26.

[6] Stanfelj N., Marlin, T. E. and MacGregor, J. F., Monitoring and diagnosing process control performance: The single-loop case, Ind. Eng. Chem. Res. 32 (1993) 301-314.

[7] Kesevan P. and Lee, J. H., Diagnostic tools for multivariable model based control systems, Ind. Eng. Chem. Res. 36 (1997) 2725-2738.

[8] Kendra S. J. and Cinar, A., Controller performance assessment by frequency domain techniques, Journal of Process Control. 7 (1997) 181-194.

[9] Kendra S. J., Basila, M. R. and Cinar, A., Intelligent process control with supervisory knowledge-based systems, Control Systems Magazine, IEEE. 14 (1994) 37-47.

[10] Kozub D., Controller performance monitoring and diagnosis: Experiences and challenges, in: CPC-V,Lake Tahoe, Ca., (1996). 83-95.

[11] Patwardhan R. S. and Shah, S. L., Issues in performance diagnostics of model-based controllers, Journal of Process Control. 12 (2002) 413-427.

[12] Kammer L. C., Gorinevsky, D. and Dumont, G. A., Semi-intrusive multivariable model invalidation, Automatica. 39 (2003) 1461-1467.

[13] Webber J. R., Gupta, Y. P. and Matapathi, S., Detection of model mismatch in dmc: Technical report Department of Chemical Engineering, Dalhousie University (2000)

[14] Ljung L., System identification theory for the user, 2nd, PTR Prentice Hall, Saddle River, N.J., (1999).

[15] Godfrey K. R., Correlation methods, Automatica. 16 (1980) 527-534.

[16] Soderstrom T. and Stoica, P., System identification, Prentice Hall, New York, (1989).

[17] Webber J. R. and Gupta, Y. P., Robust and nominal stability conditions for a simplified model predictive controller, ISA Transactions. 45 (2005) 77-86.

[18] Harris T. J., Seppala, C. T. and Desborough, L., Recent developments in controller performance monitoring and assessment techniques, in: CPC-VI,Tucson, AZ, (2001). 220-250.

[19] Harris T. J., Seppala, C. T. and Desborough, L. D., A review of performance monitoring and assessment techniques for univariate and multivariate control systems, Journal of Process Control. 9 (1999) 1-17.

[20] Hoo K. A., Piovoso, M. J., Schnelle, P. D. and Rowan, D. A., Process and controller performance monitoring: Overview with industrial applications, International Journal of Adaptive Control and Signal Processing. 17 635-662.

[21] Qin S. J., Control performance monitoring -- a review and assessment, Computers & Chemical Engineering. 23 (1998) 173-186.

[22] Huang B. and Shah, S. L., Performance assessment of control loops:Theory and applications, Springer-Verlag, Berlin, (1999).

[23] DeVries W. R. and Wu, S. M., Evaluation of process control effectiveness and diagnosis of variation in the paper basis weight via multivariate time-series analysis, IEEE Transactions on automatic control. 23 (1978) 703-705.

[24] Desborough L. and Harris, T. J., Performance assessment measures for univariate feedforward/feedback control, Can. J. Chem. Eng. 71 (1993) 605-617.

[25] Elnaggar A., Dumont, G. A. and Elshafei, A. L., New metod for delay estimation, in: Proceedings of the 29th Conference on Decision and Control,Honolulu, Hawaii, (1990). 1629-1630.

[26] Kamrunnahar M., Fisher, D. G. and Huang, B., Performance assessment and robustness analysis using an armarkov approach, Journal of Process Control. 14 (2004) 915-925.

[27] Astrom K. J., Computer control of a paper machine - an application of linear stochastic control theory, IBM Journal. (1967) 389-405.

[28] Huang B., Shaw, L. and Miller, R., Feedforward plus feedback controller perforamnce assessment of mimo systems, IEEE Transactions on Automatic Control. 8 (2000) 580-587.

[29] Eriksson P. G. and Isaksson, A. J., Some aspects of control loop performance monitoring, in: Proceedings of the Third IEEE Conference on Control Applications,Glasgow, UK, (1994). 1029-1034.

[30] Ko B. and Edgar, T. F., Assessment of achievable pi control performance for linear processess with dead time, in: American control conference,Philadelphia, PA, (1998).

[31] Grimble M. J., Restricted structure controller tuning and performance assessment, IEE Proc- Control Theory Appl. 149 (2002) 8-16.

[32] Tyler M. L. and Morari, M., Performance monitoring of control systems using likelihood methods, Automatica. 32 (1996) 1145-1162.

[33] Harris T. J., Boudreau, F. and MacGregor, J. F., Performance assessment of multivariable feedback controllers, Automatica. 32 (1996) 1505-1518.

[34] Huang B., Shah, S. L. and Kwok, E. K., Good, bad or optimal? Performance assessment of multivariable processes, Automatica. 33 (1997) 1175-1183.

[35] Ko B.-S. and Edgar, T. F., Performance assessment of multivariable feedback control systems, Automatica. 37 (2001) 899-905.

[36] Huang B. and Shah, S. L., Practical issues in multivariable feedback control performance assessment, Journal of Process Control. 8 (1998) 421-430.

[37] Huang B., Minimum variance control and performance assessment of time-variant processes, Journal of Process Control. 12 (2002) 707-719.

[38] Huang B., Performance asssessment of processes with abrupt changes of disturbances, The Canadian Journal of Chemical Engineering. 77 (1998) 1044-1054.

[39] Olaleye F., Huang, B. and Tamayo, E., Performance assessment of control loops with time-variant disturbance dynamics, Journal of Process Control. 14 (2004) 867-877.

[40] Wan S. and Huang, B., Robust performance assessment of feedback control systems, Automatica. 38 (2002) 33-46.

[41] McNabb C. A. and Qin, S. J., Projection based mimo control performance monitoring: 2--measured disturbances and setpoint changes, Journal of Process Control. 15 (2005) 89-102.

[42] McNabb C. A. and Qin, S. J., Projection based mimo control performance monitoring: I- covariance monitoring in state space, Journal of Process Control. 13 (2003) 739-757.

[43] Guay M., Dier, R., Hahn, J. and McLellan, P. J., Effect of process nonlinearity on linear quadratic regulator performance, Journal of Process Control. 15 (2005) 113-124.

[44] Hugo A., Performance assessment of dmc, in: Proceedings of the American Conrol Conference,San Diego, California, (1999). 2640-2641.

[45] Shah S. L., Patwardhan, R. and Huang, B., Multivariable controller performance analysis: Methods, applications and challenges, in: CPC VI,Tucson, Arizona, (2001). 190-207.

[46] Patwardhan R., Shah, S. L. and Qi, K. Z., Assessing the performance of model predictive controllers, The Canadian Journal of Chemical Engineering. 80 (2002) 954-966.

[47] Schafer J. and Cinar, A., Multivariable mpc system performance assessment, monitoring, and diagnosis, Journal of Process Control. 14 (2004) 113-129.

[48] Julien R. H., Foley, M. W. and Cluett, W. R., Performance assessment using a model predictive control benchmark, Journal of Process Control. 14 (2004) 441-456.

[49] Zhang Y. and Henson, M. A., Performance measure for constrained model predictive controllers, in: Proc. European Control Conf.,Karlsruhe, Germany, (1999).

[50] Ko B. and Edgar, T. F., Performance assessment of constrained model predictive control systems, AIChE. 47 (2001) 1363-1371.

[51] Quinn S. L., Harris, T. J. and Bacon, D. W., Accounting for uncertainty in control relevant statistics, Journal of Process Control. 15 (2005) 675-690.

[52] Quinn S. L., Bacon, D. W. and Harris, T. J., Assessing the precision of model predictions and other funcitons of model parameters, Can. J. Chem. Eng. 77 (1999) 723-737.

[53] Zoubir A. M. and Boashash, B., The bootstrap and its application in signal processing, Signal Processing Magazine, IEEE. 15 (1998) 56-76.

[54] Politis D. N., Computer-intensive methods in statistical analysis, Signal Processing Magazine, IEEE. 15 (1998) 39-55.

[55] Chernick, Bootstrap methods, Wiley-Interscience, (1999).

[56] Efron B. and Tibshirani, R., An introduction to the bootstrap, Chapman & Hall, New York, (1993).

[57] Urban Hjorth J. S., Computer intensive statistical methods, Chapman and Hall, London, (1994).

[58] Tibshirani R., Variance stabilization and the bootstrap, Biometrika. 75 (1988) 433-444.

[59] Carpenter J. and Bithell, J., Bootstrap confidence intervals: When, which, what? A practical guide for medical statisticians, Statistics in Medicine. 19 (2000) 1141-1164.

[60] Huang B., Shah, S. L. and Kwok, E. K., On-line control performance monitoring of mimo processes, in: Proceedings of the American Control Conference,(1995). 21-23.

[61] Webber J. R. and Gupta, Y. P., Model identification from closed-loop data, in: Canadian Chem. Eng. Conference,Halifax, (2001).

[62] Gustafsson F. and Graebe, S. F., Closed-loop performance monitoring in the presence of system changes and disturbances, Automatica. 34 (1998) 1311-1326.

[63] Huang B., Multivariable model validation in the presence of time-variant disturbance dynamics, Chemical Engineering Science. 55 (2000) 4583-4595.

[64] Huang B. and Tamayo, E. C., Model validation for industrial model predictive control systsems, Chemical Engineering Science. 55 (2000) 2315-2327.

[65] Söderström T. and Stoica, P., System identification, Prentice Hall International, London, (1989).

[66] Richalet J., Rault, A., Testud, J. L. and Papon, J., Model predictive heuristic control: Applications to industrial processes, Automatica. 14 (1978) 413-428.

[67] Morari M. and Lee, H. J., Model predictive control: Past, present and future, Computers & Chemical Engineering. 23 (1999) 667-682.

[68] Qin S. J. and Badgwell, T. A., A survey of industrial model predictive control technology, Control Engineering Practice. 11 (2003) 733-764.

[69] Richalet J., Industrial applications of model based predictive control, Automatica. 29 (1993) 1251-1274.

[70] Garcia C. E., Prett, D. M. and Morari, M., Model predictive control: Theory and practice--a survey, Automatica. 25 (1989) 335-348.

[71] Morari M. and Lee, H. J., Model predictive control: The good, the bad, and the ugly, in: Proc. of CPC-IV,South Padre Island, TX, (1991). 419-444.

[72] Muske K. R. and Rawlings, J. B., Model predictive control with linear models, AIChE. 39 (1993) 262-284.

[73] Rawlings J. B., Tutorial overview of model predictive control, IEEE Control Systems Magazine. 20 (2000) 38-52.

[74] Ricker N. L., Model-predictive control: State of the art, in: Chemical Process Control - CPC IV, fourth international conference on chemical process control,Amsterdam, (1991). 271-297.

[75] Clarke D. W., Advances in model-based predictive control, Oxford University Press, (1994).

[76] Maciejowski J. M., Predictive control with constraints, Prentice Hall, Harlow, England, (2002).

[77] Gupta Y. P., A simplified predictive control approach for handling constraints through linear programming, Computers in Industry. 21 (1993) 255-265.

[78] Cutler C. R. and Ramaker, B. L., Dynamic matrix control- a computer control algorithm, in: AIChE 86th National Meeting,Houston, TX, (1979).

[79] Garcia C. E. and Morshedi, A. M., Quadratic programming solution of dynamic matrix control, Chem. Eng. Commun. 46 (1986) 73-87.

[80] Cutler C. R., Haydel, J. J. and Morshedi, A. M., An industrial perspective on advanced control, in: AIChE Annual Meeting,Washington, DC, (1983).

[81] Grosdidier P., Froisy, B. and Hammann, M., The idcom-m controller, in: IFAC Workshop on Model-Based Process Control,(1988). 31-46.

[82] Morshedi A. M., Cuttler, C. R. and Skrovanek , T. A., Optimal solution of dynamic matrix control with linear programming techniques (ldmc), in: Proceedings of the ACC,(1985).

[83] Clarke D. W., Mohtadi, C. and Tuffs, P. S., Generalized predictive control--part i. The basic algorithm, Automatica. 23 (1987) 137-148.

[84] De Keyser R. M. C., Van de Velde, P. G. A. and Dumortier, F. A. G., A comparative study of self-adaptive long-range predictive control methods, Automatica. 24 (1988) 149-163.

[85] Garcia C. E. and Morari, M., Internal model control: 1. A unifying review and some new results, Ind. Eng. Chem. Process Des. Dev. 21 (1982) 308-323.

[86] Garcia C. E. and Morari, M., Internal model control: 3. Multivariable control law computation and tuning guidelines, Ind. Eng. Chem. Process Des. Dev. 24 (1985) 484-494.

[87] Garcia C. E. and Morari, M., Internal model control: 2. Design procedure for multivariable systems, Ind. Eng. Chem. Process Des. Dev. 24 (1985) 472-484.

[88] Seborg D. E., Edgar, T. F. and Mellichamp, D. A., Process dynamics and control, John Wiley and Sons, New York, (1989).

[89] Ljung L., Gustavsson, I. and Soderstrom, T., Identification of linear multivariable systems operating under linear feedback control, IEEE Transactions on automatic control. 19 (1974) 836-1974.

[90] Hughes M. T. G. and Noton, A. R. M., The measurement of control system characteristics by means of a cross-correlator, The Institute of Electrical Engineers. (1962) 77-83.

[91] Godfrey K. R., The theory of the correlation method of dynamic analysis and its application to industrial process nuclear power plant, Measurement and control. 2 (1969) 65-72.

[92] Davies W. D. T., Generation and properties of maximum-length sequences, Control. (1966) 302-433.

[93] Wellstead P., Non-parametric methods of system identification, Automatica. 17 (1981) 55-69.

[94] Heath W. p., Bias of indirect non-parmetric transfer function estimates for plants in closed loop, Automatica. 37 (2001) 1529-1540.

[95] Heath W. p., The variation of non-parametric estimates in closed-loop, Automatica. 39 (2003) 1849-1863.

[96] Carlstein E., The use of subseries values for estimating the variance of a general statistic from a stationary sequence, The Annals of Statistics. 14 (1986) 1171-1179.

[97] Kunsch H. R., The jackknife and the bootstrap for general stationary observations, The Annals of Statistics. 17 (1989) 1217-1241.

[98] Buhlmann P., Bootstraps for time series, Statistical Science. 27 (2002) 52-72.

[99] Politis D. N., The impact of bootstrap methods on time series analysis, Statistical Science. 18 (2003) 219-230.

[100]Kreiss J. P., Bootstrap procedures for ar(infinte) processes,(K. H. Jockel, G. Rothe and W. Sendler, eds.) Springer, Berlin, (1992), 107-113.

[101] Buhlmann P., Sieve bootstrap for time series, Bernoulli. 3 (1997) 123-148.

[102] Box G. E. P. and Jenkins, G. M., Time series analysis: Forecasting and control, Holden-Day, California, (1976).

[103] Ling W. M. and Rivera, D. E., Multivariable impulse response estimation via correlation analysis and its application to automated system identification, in: IFAC System identification,Japan, (1997). 1461-1466.

[104] Fontes C. H. O. and Embirucu, M., Multivariate correlation analysis and its applications to an industrial polymerication reactor, Computers & Chemical Engineering. 25 (2001) 191-201.

[105] Hamilton J. D., Time series analysis, Princeton University Press, (1994).

[106] Gustavsson I., Ljung, L. and Soderstrom, T., Identification of processes in closed-loop- identifiability and accuracy aspects, Automatica. 13 (1977) 59-75.