SUBSPACE PREDICTIVE CONTROL:

STABILITY AND PERFORMANCE ENHANCEMENT

by

Saba Sedghizadeh

MSc. in Electrical Engineering, Azad University, Tehran, Iran, 2002

BSc. in Electrical Engineering, Azad University, Tehran, Iran, 1998

A dissertation presented to Ryerson University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2017

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A DISSERTATION

I hereby declare that I am the sole author of this dissertation. This is a true copy of the dissertation, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this dissertation to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this dissertation by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public for the purpose of scholarly research only.

Abstract

Subspace Predictive Control: Stability and Performance Enhancement

PhD, 2017

Saba Sedghizadeh
Electrical and Computer Engineering
Ryerson University

In the absence of prior knowledge of a system, control design relies heavily on the system identification procedure. In real applications, there is an increasing demand to combine the usually time consuming system identification and modeling step with the control design procedure. Motivated by this demand, data-driven control approaches attempt to use the input-output data to design the controller directly. Subspace Predictive Control (SPC) is one popular example of these algorithms that combines Model Predictive Control (MPC) and Subspace Identification Methods (SIM). SPC instability and performance deterioration in closed-loop implementations are majorly caused by either poor tuning of SPC horizons or changes in the dynamics of the system. Stability and performance analysis of the SPC are the focus of this dissertation. We first provide the necessary and sufficient condition for SPC closed-loop stability. The results introduce SPC stability graphs that can provide the feasible prediction horizon range. Consequently, these stability constraints are included in SPC cost function optimization to provide a new method for determining the SPC horizons. The novel SPC horizon selection enhances the closed-loop performance effectively. Note that time-delay estimation and order selection in system modeling have been a challenging step in applications and industry. Here, we propose a new approach denoted by RE-based TDE that simultaneously and efficiently estimates the time-delay for the SIM framework. In addition, we use the recently developed MSEE approach for estimating the system order. Moreover, we propose an artificial intelligence approach denoted by Particle Swarm Optimization Based Fuzzy Gain-Scheduled SPC (PSO-based FGS-SPC). The method overcomes the issue of on-line adaptation of SPC gains for systems with variable dynamics in the presence of the noisy data. The approach eliminates existing tuning problem of controller gain ranges in FGS and updates the SPC gains with no need to apply any external persistently excitation signals. As a result, PSO-based FGS-SPC provides a time efficient control strategy with fast and robust tracking performance compared to conventional and state of the art methods.

Contents

List of Tables

List of Figures

List of Acronyms

CGRs	Controller Gain Ranges
DC	Direct Current
DMC	Dynamic Matrix Control
FGS	Fuzzy Gain Scheduling
FGS-PID	Fuzzy Gain-Scheduled Proportional Integral Derivative
FGS-SPC	Fuzzy Gain-Scheduled Subspace Predictive Control
FR	Feasible Range
GA	Genetic Algorithm
GPC	Generalized Predictive Control
GS	Gain Scheduling
I/O	Input-Output
IESS	Integrator Embedded State-space
LQG	Linear Quadratic Gaussian
LTI	Linear Time Invariant
MIMO	Multi-Input Multi-Output
MOESP	Multivariate Output Error State-Space
MPC	Model Predictive Control
MSEE	Mean Squared Eigenvector Error
N4SID	Numerical Subspace State-Space System Identification
NB	Negative Big
NM	Negative Medium
NS	Negative Small
PB	Positive Big
PD	Proportional-Derivative

PE	Persistently Exciting
PEM	Prediction Error Method
PID	Proportional Integral Derivative
PM	Positive Medium
PRBS	Pseudo Random Binary Signal
PS	Positive Small
PSO	Particle Swarm Optimization
QMPC	Quadratic Dynamic Control
QP	Quadratic Programing
RE	Reconstruction Error
SFP	Shortest-Feasible-Prediction-Horizon
SIM	Subspace Identification Methods
SISO	Single-Input Single-Output
SNR	Signal to Noise Ratio
SP	Subspace Predictor
SPC	Subspace Predictive Control
SVD	Singular Value Decomposition
TDE	Time Delay Estimation
ZZ	Zero

List of Notations

$x(k)$	System state variable
$u(k)$	System input
$y(k)$	System output
$\zeta(k)$	Innovation sequence
$r(k)$	Reference signal
$d(k)$	Disturbance signal
$e(k)$	Tracking error
$\hat{y}(k)$	Predicted output
A, B, C, D	State-space system matrices
K	Kalman gain matrix
k	Sample step
n	System order
m	Number of system inputs
l	Number of system outputs
p	Predictor past horizon
f	Predictor future horizon
N	Length of available I/O data
M	Order of subspace predictor
U_p, U_f	Past and future input data block Hankel matrices
Y_p, Y_f	Past and future output data block Hankel matrices
Z_p, Z_f	Past and future noise block Hankel matrices
X_p, X_f	Past and future state variable matrices
W_p	Past I/O data block Hankel matrix
$\hat{Y}_f$	Predicted future output data block Hankel matrix

Γ_M	Extended observability matrix
H_M^d, H_M^s	Lower triangular block Toeplitz matrices (deterministic and stochastic)
Δ_M^d, Δ_M^s	Reversed extended controllability matrices (deterministic and stochastic)
u_p, u_f	Past and future input data vectors
y_p, y_f	Past and future output data vectors
w_p	Past I/O data vector
t_d	Time-Delay
$\hat{y}_f$	Predicted future output data vector
t	Iteration step of PSO algorithm
$p(t)$	Particle position
$pb(t)$	Particle best position
pg	Global best position
r_f	Future reference signal vector
L_u, L_w	Subspace linear predictor coefficient matrices
$\tilde{L}_u, \tilde{L}_w$	Truncated subspace linear predictor coefficient matrices
$K_e, K_{\Delta w}$	SPC gain matrices
$\tilde{K}_e, \tilde{K}_{\Delta w}$	Truncated SPC gain matrices
$\tilde{k}_e, \tilde{k}_{\Delta w}$	Truncated SPC gain vectors
$\bar{k}_e, \bar{k}_{\Delta w}$	Normalized SPC gain vectors
F_1, F_2	Auxiliary matrices
J	SPC cost function
Q, R	SPC weighting matrices
N_c	Control Horizon
N_p	Prediction Horizon
N_d	Sample-Delay
N_{EP}	Efficient prediction horizon
N_{EC}	Efficient control horizon
N_{SFP}	Shortest feasible prediction horizon
q^{-1}	Backward shift operators
G_{ol}	Open-loop transfer function
G_{cl}	Closed-loop transfer function
K_{co}	DC-gain of compound system

K_{ol}	DC-gain of open-loop system
e_{RE}	Reconstruction error

Chapter 1

Introduction

1.1 Dissertation Objectives

Two fundamental steps in developing control systems are: (i) obtaining model of the system, and (ii) designing a controller for the system. Both steps have significant impacts on the closed-loop stability and performance of the implemented control system.

General approaches for the first step are *system modeling* and *system identification.* System modeling uses physics laws such as, Newton's law [1], Ohm's law [2] and thermodynamic principles [3] to obtain an appropriate mathematical description of the system. This approach is well-developed in control system design and there are many successful implementations of such technique. Nevertheless, system modeling is not applicable for complex and multivariate systems, due to expensive, time consuming and inaccurate performance of such dynamic systems. System identification is another approach to obtain the system model. It develops mathematical model of the system by using experimental data [4]. There are many applications of system identification in control with substantial overlap in other areas such as signal processing, communication and statistics. Some of the identification methods have *black-box* approach to the identified system which is good for automatic identification, where no additional information about the identified system is available. However, the input-output (I/O) data may be corrupted and the pure black-box approach may lead to a model, which is significantly deviated from the real system. There is often a priori information, which should be exploited to increase the quality of the identification and to ensure that the identified model is close to some expectations. These approaches are called *gray-box* methods [5].

In the second step, a controller is designed based on the system model developed in the first step and the specifications of the closed-loop system performance. There are various types of controllers from traditional Proportional-Integral-Derivative (PID) controllers, most popular controller in the industry, to the advanced sophisticated modern control methods. Among them Predictive Control Method is a class of control algorithms that determines the control law from a predictive model of the system by minimizing an appropriate objective function. There is a *model-based* approach to predictive control, which is called *Model-based Predictive Control* (MBPC) or *Model Predictive Control* (MPC) [6–10]. MPC was introduced in early 80's which calculates dynamic model of the system from I/O data using system identification techniques and uses the model to construct the *predictor matrices*. MPC utilizes on-line optimization of a linear or quadratic objective function subject to linear constraints over a finite prediction horizon [7,11,12]. The controller uses different system models, such as impulse response or step response series [13,14], transfer function models [15,16] and state-space realizations [12,17] in the presence of both soft and hard constraints.

On the other hand, it has been found in late 90's that these predictor matrices can be obtained directly from the experimental I/O data [18,19]. This approach is called *data-driven* or *model-free* predictive control, which is based on an appropriate combination of MPC with Subspace Identification Methods (SIM) [18–20]. In data-driven approach the predictor matrices are determined directly from the experimental I/O data by using subspace predictor matrices which eliminates the intermediate parametric model identification step [18,19]. The resultant methodology is called *Data-driven Subspace Predictive Control* or briefly *Subspace Predictive Control* (SPC) in the literature [18], [21]. In SPC methodology the system identification and the controller design steps are combined and implemented in a single-step which is calculating the subspace predictor. Some features of SPC, such as no pre-assumptions about system model, calculation of predictor matrices without iteration and no need to solving Diophantine equation are some of the advantages of SPC in practical applications [21,22]. With increasing popularity of MPC and SPC in industrial applications [23] such as, chemical engineering [24–26], power systems [27–29], smart grids and buildings [30,31], network control systems [32,33], vehicle control [34], their closed-loop stability and performance have become significant and controversial issues in predictive control [35–38]. This dissertation focuses on studying the SPC methodology and overcoming some of its well-known issues.

SPC Stability

Due to the finite horizons formulation in MPC and SPC, stability and feasibility are not guaranteed for all time instants. Stability issue of MPC is studied in several papers in the literature, an exhaustive review of stability solutions for MPC is provided in [39]. In [40] and [41] closed-loop stability criterion is presented for extended predictive control by taking into account the prediction horizon value and the reference trajectory dynamics. Necessary and sufficient conditions of infinite horizon Generalized Predictive Control (GPC) GPC$^{\infty}$ is presented in [42] with no need to solve a Lyapunov equation. Stability conditions for constrained GPC$^{\infty}$ and constrained receding horizon predictive control are presented in [43] by imposing terminal constraints on inputs and outputs over some constraint horizon and checking the monotonicity of cost function. The MPC stability is investigated in [44] by checking the monotonicity of cost function for a nonlinear predictor based on neural network. In [45] several criteria are provided to guarantee the closed-loop stability of GPC in the field of self-tuning control. However, all these MPC closed-loop stability strategies have been developed with assumption that the system dynamic model or some prior information, such as open-loop settling time, rise time, time-delay are available and all state variables can be measured [46–49]. Since SPC is a model-free approach, the existing model-based techniques for tuning and stability are not applicable for SPC, and closed-loop stability of SPC is still an open problem. Therefore, stability analysis of the SPC is the main research objective of this dissertation.

SPC Performance

Performance monitoring of a closed-loop system has three steps: First, an appropriate benchmark is defined for comparison and to determine the controller capability. Second, the controller operation by proper monitoring statistics is assessed [50, 51]. Third, possible performance degradation of the controller is detected. The main focus of the existing works on performance monitoring of SPC concentrates on the first and second steps [36, 52], and the third step is not addressed adequately. Generally, there are some internal and external factors that cause performance deterioration in SPC [53]. Internal factors include changes in real process which affect the subspace predictor matrices in SPC, changes in the controller constraints, and poor tuning of the SPC parameters. External factors include noisy data, disturbances, length of data and malfunction of sensors/actuators. This dissertation focuses on changes in real process in the presence of noisy data and tuning of the SPC parameters to achieve an efficient performance.

SPC Tuning Parameters

MPC and SPC have same cost functions and tuning parameters, which include prediction horizon, control horizon and weighting matrices to penalty the tracking error, control signal and its variation. Appropriate choice of these parameters can significantly influence the closed-loop stability and performance. There are extensive studies in the literature that provide several tuning strategies for MPC [48, 54–58]. There exist some attempts to improve the performance and computational efficiency of MPC algorithms for fast dynamic applications by selecting small horizons. In [59] the cost function with a single horizon is considered and the optimality of control law with horizon one is discussed. In this particular application, however, the inputs are limited to a finite set of data and analysis is for a very restricted specific application. Moreover, a small deviation from the actual system can jeopardize the stability and optimality of the system. In [60] a nonlinear MPC formulation of I/O linearizion in continuous time is given for single-input single-output (SISO) systems, and shown that an I/O linearizing state feedback is a shortest-prediction-horizon MPC law. However, the method is suffering from instability for non-minimum-phase systems. This issue is addressed in [38] through the use of Lyapunov inequality constraints. All the above mentioned methods are restricted to special cases and cannot be generalize to obtain optimum horizon. Consequently, the second research objective of this dissertation is determining the feasible range of SPC horizons based on model-free algorithms to guarantee the closed-loop stability, and to achieve an optimum feasible performance by obtaining the optimum horizons.

SPC Gains Updating

In earlier SPC algorithms, the subspace predictor matrices are calculated off-line before the actual implementation by using open-loop I/O data [18, 21], as well as closed-loop I/O data [61, 62]. However, these approaches have not adequately addressed issue of on-line controller adaptation for time-varying systems. In application of SPC, due to existence of nonlinearities and parameter uncertainties in real processes, the input signals have to be persistently exciting (PE) the system by randomly generated signals. This feature enables SPC to obtain sufficient information from new I/O data and to predict the system outputs accurately. However, because of the high computational load and disruption in system operation, the all times persistent excitation should be avoided, especially in the steady-state mode. Therefore, much attention is given to replacing this time consuming updating process with an efficient algorithm [63]. There are several works that propose different adaptive algorithms to combine with SPC. For instance, in [64] the excitation method is illustrated by formulating additional constraints to the optimization problem, however, this approach results in solving non-convex optimization problem to compute the control signal at each time step. A similar approach is also found in [65] to assure PE in the input signal. Moreover, [66] presents a method to update the predictor in recursive manner by considering an exponential forgetting factor. Another recursive SPC technique that is based on the orthogonal Givens rotations and variable forgetting factors is demonstrated in [67] and [68]. In [68] a strategy to determine the sufficient time to apply the excitation signal to the system is proposed by considering the prediction error. A time-varying forgetting factor based strategy is presented in [69] to apply in an adaptive manner to the SPC. In [70] an extra term has been added to the cost function to ensure the PE, and to keep the optimization problem convex and quadratic, however, it degrades the control performance. Although, there have been some attempts to reduce the destructive effects of applying the all-time PE signals, but non of them could completely eliminate the requirement of PE in SPC. Therefore, providing an efficient method to eliminate the requirement of applying PE signals to update the SPC gains is the third research objective of this dissertation.

Time-delay and Order Selection in SIM and SPC

SIM and SPC are *black-box* data-driven approaches, which do not need to have prior knowledge about the system. In SPC, the subspace predictor matrices are derived directly from I/O data. This feature of SPC makes it appropriate for automatic control purposes. However, when I/O data is corrupted by noise, the pure black-box data-driven approach may lead to an incorrect result in prediction and deteriorate the SPC performance. In this scenario, incorporating some a priori information can help to appropriately select the SPC parameters and increase the quality of the prediction and performance. Some of the a priori information, which would be useful to incorporate are the system order, sign of DC-gain and time-delay. In SIM framework these parameters can be determined from the subspace predictor matrices, which are derived via I/O data block-Hankel matrices. However, estimation of these parameters from noisy I/O data is a tricky task. There are several well known order selection methods in the literature which are based on information criteria such as, Akaike's Information Criterion (AIC) [71] and Minimum Description Length (MDL) [72]. However, it has been shown that both AIC and MDL tend to overestimate the system order by increasing the Signal-to-Noise Ratio (SNR) [73]. Furthermore, MDL tends to underestimate the system order at low SNR. Moreover, most of the existing time-delay estimation techniques are threshold based methods such as, Cumulative Sum (CUSUM) method [74], Separating Frequency

Method [75] and Subspace based method [76], which are not applicable for low SNR noisy data. Therefore, two new statistical approaches are presented in this dissertation to estimate the system order and time-delay, and the advantages of the proposed methods are shown over the existing methods.

1.2 Dissertation Outline

The dissertation structure are as follows:

Technical background of the dissertation is provided in Chapter 2. Classical System Identification, Subspace System Identification, Model-based and Data-driven approaches of Predictive Control are reviewed in this Chapter. Model Predictive Control (MPC) and Subspace Predictive Control (SPC) algorithms are briefly discussed. Last part of the chapter focuses on a general review of Fuzzy Gain-Scheduling (FGS) and Particle Swarm Optimization (PSO) techniques.

Chapter 3 provides SPC closed-loop stability analysis, and the necessary and sufficient condition for SPC closed-loop stability of open-loop stable systems is derived based on a model-free approach. Consequently, SPC stability graph is introduced, and a model-free technique is provided to find the feasible range (FR) of prediction horizon that guarantees the closed-loop stability. Next, a technique is introduced to select the efficient control horizon and prediction horizon by optimizing the SPC cost function.

Chapter 4 addresses issues of SPC on-line adaption for time-varying systems, and introduces an artificial intelligence based approach to update the SPC gains. A PSO-based FGS-SPC method is proposed to eliminate the requirement of applying persistently exciting signal to system and their drawback. Moreover, existing issue of FGS in determining the controller gain ranges (CGRs) in PSO-based FGS technique is addressed and solved. Therefore, PSO-based FGS itself can be considered as a separate package for applying to any system which needs Gain-Scheduling (GS) technique. The formulation and simulation results are given in detail. The proposed method is compared with FGS-SPC and SPC method to illustrate its efficiency and performance in the presence of noisy data.

In Chapter 5, the issue of black-box data-driven SPC approach in the presence of noisy data is discussed. The Chapter shows the benefits of incorporating some prior knowledge such as system order and time-delay in SIM and SPC. Mean Squared Eigenvalue Error (MSEE) method is implemented for order selection, and for time-delay estimation the Reconstruction Error Based

(RE-based) method is introduced. Simulation results show the superiority of these methods over the existing methods.

1.3 Dissertation Contributions

The contributions of this dissertation are summarized as follows:

- A necessary and sufficient condition for SPC closed-loop stability is provided for open-loop stable systems based on SPC gains and SPC tuning parameters. Consequently, SPC stability graph is introduced to determine the feasibility range of prediction horizon (FR of N_p) that guarantees SPC closed-loop stability. The method is applicable to non-minimum phase systems.

- Efficient control horizon (N_{EC}) is determined by minimizing the dimension of subspace predictor matrix. Efficient prediction horizon (N_{EP}) is then provided by minimizing the SPC cost function to ensure the closed-loop stability holds in the presence of noise. Moreover, this optimal selection of SPC horizon length decreases the computational requirements of the SPC optimization algorithm without jeopardizing the closed-loop stability and performance.

- PSO-based FGS-SPC method is proposed to optimally update the SPC gains for time-varying systems in the presence of constraints and noise. The proposed method eliminates the requirement of applying PE signals to update the subspace predictor matrices, and reduces the computational complexity load of the gain updating procedure in SPC.

- The proposed PSO-based FGS procedure by itself is an optimal technique to tune the CGRs in FGS procedure, which is applicable for any system that uses FGS method. This technique overcomes the problem of calculating the CGRs in FGS algorithm, by optimally calculating the CGRs via minimizing the SPC cost function.

- Novel data-driven method for order selection and new time-delay estimation method are presented in SIM framework. RE-based TDE is used for time-delay estimation and MSEE is utilized for order selection. These data-driven approaches can also be utilized for SPC in case the system order or time-delay are required.

Chapter 2

Background

2.1 System Identification Methods

System identification is a well-known procedure to obtain model of the system by using experimental data. There are comprehensive literature and many applications for system identification in different fields such as control systems, chemical and petroleum engineering, signal processing, statistics and more. In the area of control systems following publications by Ljung [4], Sinha and Kuszta [77], Soderstrom and Stoica [78] can be mentioned. Generally, system identification for linear systems has three main steps:

Experiment Design and Data Acquisition

In this step the I/O data is collected by exciting the system with appropriate persistently exciting (PE) input signals and choosing a good sampling rate around the operating point. These PE signals must sufficiently rich to excite the system so that all important aspects of the system behavior appear in the system's output. Gaussian white noise and Pseudo Random Binary Signals (PRBS) are two of the most popular PE signals in system identification. Data collecting procedure can be done in open-loop or closed-loop conditions by considering the safety and stability of the system. Sometimes, it needs to have some physical knowledge or a priori information about the system to enhance the identification [79], [80].

Model Structure Determination

After gathering the enough data the suitable dynamic model structure is selected for the system based on the chosen control strategy. These models include state-space models, transfer functions and time series models for parametric methods [81–83]. Also, it could be a non-parametric models such as impulse responses, frequency responses etc. [84–86]. One of the most challenging problems in fixing the good model structure is to find the system order and existing time-delay from I/O data.

Parameter Estimation and Validation

After choosing the model structure its parameters need to be estimated by one of the well-known system identification techniques. Typical techniques include the Prediction Error Method (PEM) [4, 87] which uses Least Squares and Maximum Likelihood techniques. The Instrumental Variable Method (IVM) which uses correlation algorithms [88, 89]. The Subspace Identification Methods (SIM) which use experimental data to estimate the state-space model of the system [19, 20, 90]. Finally, the model validation should be done to ensure that the estimated model can describe the system appropriately. In this step some techniques such as spectral and residual analysis may be hired to find the degree of confidence of the model. The whole procedure needs to be tried again, if the results are unsatisfactory. In addition, there are several criterion to select the best model among the several estimated models, such as, Akaike Information Criterion (AIC) [91], Baysian Information Criterion (BIC) [92], Minimum Description Length (MDL) [72] and Minimum Description Complexity (MDC) [93, 94].

2.1.1 Classical Approach

While this dissertation only concentrates on the Subspace approach of system identification which are used for SPC, here we provide a brief review on Classical approaches of system identification that are historically before Subspace approach.

Classical system identification theory has been pioneered by Astrom and Bohlin [81] in 1960's by introducing the PEM. PEM is based on parameter optimization which uses the maximum likelihood framework. In PEM prediction error is defined as difference between the measured output $y(k)$ and the one-step ahead predicted output $\hat{y}(k)$. The prediction error is used to create a cost function, then the cost function is optimized with respect to the parameters of the model. Hence, selecting model is the most crucial part of the PEM.

The PEM was further extended by Ljung [4] who defined different classes of transfer-function-based models. In some of these models the prediction error is linear in the model parameters, which allows for simple optimal solution by using the Least Squares method. However, for most of the cases the problem is nonlinear, so a nonlinear optimization needs to be utilized. Some of the crucial problems in PEM framework are selecting the model class, determining order of the polynomials and choosing starting value of the model parameters, which are troublesome procedures especially in multi-input multi-output (MIMO) case. Procedure of the PEM can be summarized in four main steps [19]:

Model Structure Selection

Consider the *true* system represented as follows,

$$y(k) \quad = \quad \bar{G}_p(q^{-1})u(k) + d(k) \tag{2.1}$$

where k is the k-th sampling time, $u(k)$ is the input, $y(k)$ is the output, $\bar{G}_p(q^{-1})$ is the *true* system, q^{-1} is the backward shift operator, $d(k)$ is the output disturbance. A general stable linear time invariant (LTI) system model has following structure,

$$y(k) \quad = \quad G_p(q^{-1};\theta)u(k) + G_w(q^{-1};\theta)w(k) \tag{2.2}$$

where $G_p(q^{-1};\theta)$ is the system model $G_w(q^{-1};\theta)$ is the noise model, $w(k)$ is white noise with zero mean and variance σ^2, and θ is a set of all unknown parameters other than the noise variance. The problem is to fit the LTI model in (2.2) to a given I/O data set $\{[u(k), y(k)], k = 1, ..., N\}$.

The most commonly used class of model structures in the PEM can be described by following equation,

$$A(q^{-1})y(k) \quad = \quad \frac{B(q^{-1})}{F(q^{-1})}u(k) + \frac{C(q^{-1})}{D(q^{-1})}w(k) \tag{2.3}$$

where

$$A(q^{-1}) \quad = \quad 1 + a_1 q^{-1} + \cdots + a_{na} q^{-na} \tag{2.4}$$

$$B(q^{-1}) \quad = \quad b_1 q^{-1} + \cdots + b_{nb} q^{-nb} \tag{2.5}$$

$$C(q^{-1}) \quad = \quad 1 + c_1 q^{-1} + \cdots + c_{nc} q^{-nc} \tag{2.6}$$

$$D(q^{-1}) \quad = \quad 1 + d_1 q^{-1} + \cdots + d_{nd} q^{-nd} \tag{2.7}$$

$$F(q^{-1}) \quad = \quad 1 + f_1 q^{-1} + \cdots + f_{nf} q^{-nf} \tag{2.8}$$

the parameter to be estimated is,

$$\theta = [a_1, \dots, a_{na}, b_1, \dots, b_{nb}, c_1, \dots, c_{nc}, d_1, \dots, d_{nd}, f_1, \dots, f_{nf}]^T \tag{2.9}$$

Different model structures are obtained from the PEM model in (2.3) by considering some assumptions:

- ***Auto-Regressive with Exogenous Input Model* (ARX):** By considering $C(q^{-1}) = D(q^{-1}) = F(q^{-1}) = 1$ in the PEM model (2.3), the ARX model can be obtained,

$$A(q^{-1})y(k) = B(q^{-1})u(k) + w(k) \tag{2.10}$$

ARX model is a commonly used structure in system identification due to its simple and easily estimated structure.

- ***Auto-Regressive Moving Average with Exogenous Input Model* (ARMAX):** By removing the restriction from the numerator of the disturbance model in the ARX model structure ARMAX model can be obtained,

$$A(q^{-1})y(k) = B(q^{-1})u(k) + C(q^{-1})w(k) \tag{2.11}$$

The ARMAX model is useful in the design of Kalman filter and Generalized Predictive Control (GPC).

- ***Box-Jenkins Model* (BJ):** In BJ the model structure different denominators are considered for plant and disturbance models, unlike the ARMAX model,

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + \frac{C(q^{-1})}{D(q^{-1})}w(k) \tag{2.12}$$

The BJ model can be useful when the disturbance has a completely different model from the plant.

- ***Output Error Model* (OE):** OE model is a special case of PEM if the estimation of the disturbance is not of interest,

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + w(k) \tag{2.13}$$

The OE model is a good option of model in practice due to have fewer parameters.

- ***State-space Model*:** Generally, in PEM an innovation form of the state-space model is used,

$$
\begin{aligned}
x(k+1) &= A(\theta)x(k) + B(\theta)u(k) + K(\theta)\zeta(k) && (2.14)\\
y(k) &= C(\theta)x(k) + D(\theta)u(k) + \zeta(k) && (2.15)
\end{aligned}
$$

where A, B, C and D are system matrices, K is the Kalman predictor gain, and $\zeta(k)$ is innovation sequence. This state-space model is more suitable for MIMO systems, and it is transferable to the general form of the model in (2.2) by using following equations,

$$
\begin{aligned}
G_p(q^{-1};\theta) &= C(\theta)[qI - A(\theta)]^{-1}B(\theta) && (2.16)\\
G_d(q^{-1};\theta) &= I + C(\theta)[qI - A(\theta)]^{-1}K(\theta) && (2.17)
\end{aligned}
$$

Optimal Predictor Determination

In PEM the prediction error is defined as follows,

$$
\epsilon(k;\theta) = y(k) - \hat{y}(k;\theta) \tag{2.18}
$$

where $\epsilon(k;\theta)$ is the one-step ahead prediction error, $y(k)$ is the current output and $\hat{y}(k;\theta)$ is the one-step predicted output. The optimal one-step ahead predictor $\hat{y}(k;\theta)$ and the one-step prediction error $\epsilon(k;\theta)$ are given as follows by considering (2.1) and (2.2),

$$
\begin{aligned}
\hat{y}(k;\theta) &= G_d^{-1}(q^{-1};\theta)G_p(q^{-1};\theta)u(k) + [I - G_p^{-1}(q^{-1};\theta)]y(k) && (2.19)\\
\epsilon(k;\theta) &= G_d^{-1}(q^{-1};\theta)[\bar{G}_p(q^{-1}) - G_p(q^{-1};\theta)u(k)] + G_d^{-1}(q^{-1};\theta)d(k) && (2.20)
\end{aligned}
$$

For a state-space model in equations (2.14) and (2.15) in innovation form, the optimal predictor and the prediction error $\epsilon(k;\theta)$ are computed as follows,

$$
\begin{aligned}
\hat{x}(k+1;\theta) &= [A(\theta) - K(\theta)C(\theta)]\hat{x}(k;\theta) + B(\theta)u(k) + K(\theta)y(k) && (2.21)\\
\hat{y}(k;\theta) &= C(\theta)\hat{x}(k;\theta) + D(\theta)y(k) && (2.22)\\
\epsilon(k;\theta) &= -C(\theta)\hat{x}(k;\theta) - D(\theta)u(k) + y(k) && (2.23)
\end{aligned}
$$

Prediction Error Collection

From the given experimental I/O data $\{[u(k), y(k)], k = 1, ..., N\}$ the predictions and the corresponding sequence of prediction errors are obtained as,

$$\epsilon(k;\theta) = y(k) - \hat{y}(k;\theta), \quad k = 1, ..., N \tag{2.24}$$

Cost Function Definition

To minimize the prediction error $\{\epsilon(k;\theta),\ k = 1, ..., N\}$ in PEM the trace of sample covariance matrix of prediction errors is selected as a cost function, $J_N(\theta)$, to estimate the optimal $\hat{\theta}$,

$$\min_{\theta} \quad J_N(\theta) = trace[\frac{1}{N}\sum_{i=1}^{N} \epsilon(k;\theta)\epsilon^T(k;\theta)] \tag{2.25}$$

which is generally a complicated function of system parameters, and needs to be solved by iterative descent methods, which may get stuck in local minima. In the case of using ARMAX models the PEM reduces to a nonlinear optimization problem. Block diagram of PEM is shown in Figure 2.1 [19].

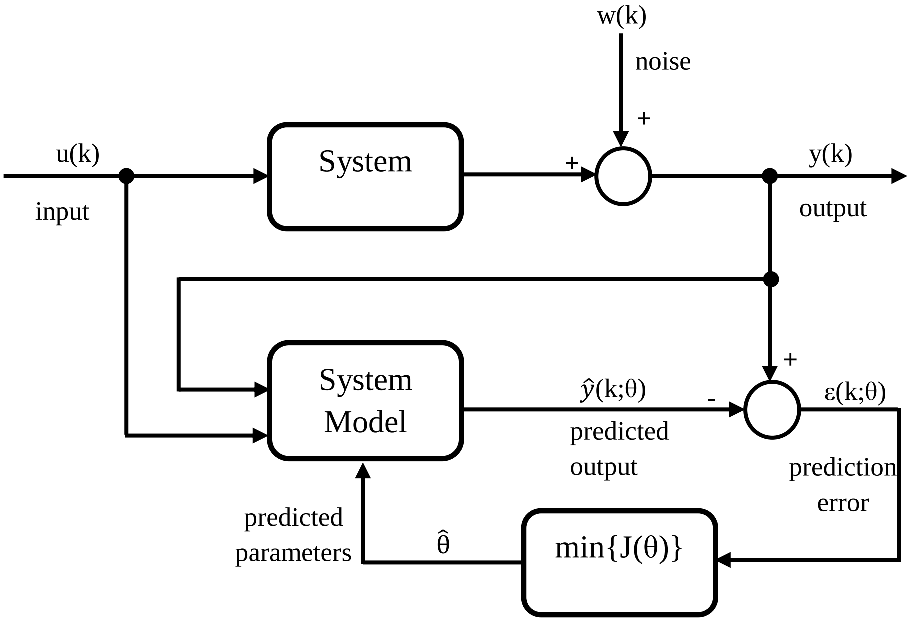

Figure 2.1: General block diagram of the Prediction Error Method

2.1.2 Subspace Approach

Foundation of the system identification based on state-space model started in 1960's with solving the deterministic state-space realization problem by Ho and Kalman [95] for the first time. They show that how a state-space model can be obtained from the impulse response coefficients by forming a certain Hankel matrix. The realization algorithm is based on the property that the Hankel matrix, which is build from impulse response parameters is equal to the product of the extended observability and controllability matrices. However, the approach is only allowed for the determination of a deterministic state-space model from impulse responses. Stochastic state-space realization was further introduced by Akaike [96].

Later in 1990's the Subspace Identification Method (SIM) algorithms were developed [20,97] via extracting the system matrices from I/O data by using modern linear algebra tools. In SIM framework the state vector $x(k)$ is estimated from I/O data matrices by utilizing a projection algorithm, and linear algebra tools such as, Singular Value Decomposition (SVD) and QR-factorization. By having the state $x(k)$, input $u(k)$ and output $y(k)$ vectors the state-space matrices A, B, C and D can be estimated using Least Squares technique. Some of the important features of SIM can be listed as follows:

1- SIM framework is based on computational tools such as, SVD and QR-factorization, which makes it intrinsically robust from a numerical point of view.

2- No need to optimize an explicit cost function allowed SIM to avoid local minima and convergence problems.

3- Theoretical and experimental evidence have shown that some SIM algorithms are close to optimal, in the sense that they approach the Cramer-Rao bound.

4- SIM is a non-iterative procedure and it is more reliable to deal with MIMO systems.

An overview of SIM algorithms is given in this section.

System Description

There are several state-space representation forms of a system. One of the most commonly used form in system identification is the innovation form. A discrete LTI system of order n can be described in state-space innovation form as,

$$x(k+1) = Ax(k) + Bu(k) + K\zeta(k) \tag{2.26}$$

$$y(k) = Cx(k) + Du(k) + \zeta(k) \tag{2.27}$$

where k is the k-th sampling time, $u(k) \in \mathbb{R}^m$ is the input, $y(k) \in \mathbb{R}^l$ is the output, $x(k) \in \mathbb{R}^n$ is the state of the system, and $\zeta(k) \in \mathbb{R}^l$ is an unknown innovation sequence with $E[\zeta(k)\zeta^T(k)] = S \in \mathbb{R}^{l\times l}$ as the innovation covariance matrix. The matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{l\times n}$, $D \in \mathbb{R}^{l\times m}$ are state-space matrices that describe the system, and $K \in \mathbb{R}^{n\times l}$ is the Kalman gain of the stable system. $\{A, C\}$ and $\{A, B\}$ are assumed to be observable and controllable, respectively.

There is also another form of state-space model which has been used in subspace identification literature as below,

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{2.28}$$

$$y(k) = Cx(k) + Du(k) + v(k) \tag{2.29}$$

with

$$\mathbf{E}\left[\begin{pmatrix} w(k) \\ v(k) \end{pmatrix}\begin{pmatrix} w^T(l) & v^T(l) \end{pmatrix}\right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix}\delta_{kl} \geq 0 \tag{2.30}$$

where δ_{kl} is the Kronecker delta, $Q \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{n\times l}$ and $R \in \mathbb{R}^{l\times l}$. The process noise $w(k) \in \mathbb{R}^n$ and the measurement noise $v(k) \in \mathbb{R}^l$ are considered as the unobserved, Gaussian distributed, zero mean, white noise vector sequences. Note that the process noise represents the accuracy and time-lag in the estimated value, and the measurement noise represents noise characteristics of the sensor. $\{A, C\}$ and $\{A, [\begin{array}{cc} B & Q^{1/2} \end{array}]\}$ are assumed to be observable and controllable, respectively. It can be shown that the model in equations (2.28) and (2.29) is also convertible to innovation form in (2.26) and (2.27) [98].

Subspace Equations

Suppose that measurements data of input $u(k)$ and output $y(k)$ are available for $k \in \{1, 2, \ldots, N\}$, where N is large enough ($N \rightarrow \infty$). In addition, to have a consistent identification we assume that, the input $u(k)$ is PE signal and it is uncorrelated with $\zeta(k)$, [20, 98]. The concept of persistency of excitation is defined in [20, 98, 99] as follows,

Definition 2.1.1. *The input signal $u(k)$ is PE of order i if the following limit exists,*

$$r_u(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} u(k+\tau)u^T(k) \tag{2.31}$$

and the following input covariance matrix is positive definite.

$$R_u(i) = \begin{bmatrix} r_u(0) & r_u(1) & \dots & r_u(i-1) \\ r_u(-1) & r_u(0) & \dots & r_u(i-2) \\ \vdots & \vdots & \vdots & \vdots \\ r_u(1-i) & \dots & \dots & r_u(0) \end{bmatrix} \tag{2.32}$$

In order to obtain the subspace equations the state-space representation in equations (2.26) and (2.27) is reformulated. For time instant k we have,

$$x(k+1) \;=\; Ax(k) + Bu(k) + K\zeta(k) \tag{2.33}$$

for time instant $k+1$ and by substituting (2.33) we have,

$$\begin{aligned} x(k+2) &= Ax(k+1) + Bu(k+1) + K\zeta(k+1) && (2.34)\\ &= A(Ax(k) + Bu(k) + K\zeta(k)) + Bu(k+1) + K\zeta(k+1) && (2.35)\\ &= A^2x(k) + \begin{bmatrix} AB & B \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} + \begin{bmatrix} AK & K \end{bmatrix} \begin{bmatrix} \zeta(k) \\ \zeta(k+1) \end{bmatrix} && (2.36) \end{aligned}$$

by continuing this procedure for time instant $k+M-1$ can be written,

$$\begin{aligned} x(k+M) = A^M x(k) + \begin{bmatrix} A^{M-1}B & A^{M-2}B & \dots & B \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+M-1) \end{bmatrix} \\ + \begin{bmatrix} A^{M-1}K & A^{M-2}K & \dots & K \end{bmatrix} \begin{bmatrix} \zeta(k) \\ \zeta(k+1) \\ \vdots \\ \zeta(k+M-1) \end{bmatrix} \end{aligned} \tag{2.37}$$

Therefore, by collecting the state variables in a vector as below,

$$[x(k+M)\ \ x(k+M+1)\ \ \ldots\ \ x(k+N-(M-1))] \tag{2.38}$$

following equation is obtained,

$$\begin{aligned}
&\begin{bmatrix} x(k+M) & x(k+M+1) & \ldots & x(k+N-(M-1)) \end{bmatrix} = A^M \begin{bmatrix} x(k) & x(k+1) & \ldots & x(k+M-1) \end{bmatrix} \\
&+ \begin{bmatrix} A^{M-1}B & A^{M-2}B & \ldots & B \end{bmatrix} \begin{bmatrix} u(k) & u(k+1) & \ldots & u(k+N-(M-1)) \\ u(k+1) & u(k+2) & \ldots & u(k+N-(M-2)) \\ \vdots & \vdots & \vdots & \vdots \\ u(k+M-1) & u(k+M) & \ldots & u(k+N) \end{bmatrix} \\
&+ \begin{bmatrix} A^{M-1}K & A^{M-2}K & \ldots & K \end{bmatrix} \begin{bmatrix} \zeta(k) & \zeta(k+1) & \ldots & \zeta(k+N-(M-1)) \\ \zeta(k+1) & \zeta(k+2) & \ldots & \zeta(k+N-(M-2)) \\ \vdots & \vdots & \vdots & \vdots \\ \zeta(k+M-1) & \zeta(k+M) & \ldots & \zeta(k+N) \end{bmatrix}
\end{aligned} \tag{2.39}$$

By applying the same recursive substitution for (2.27) following equations are obtained. For time instant k we have,

$$y(k) \quad = \quad Cx(k) + Du(k) + \zeta(k) \tag{2.40}$$

for time instant $k+1$ and by substituting (2.26) we have,

$$\begin{aligned}
y(k+1) &= Cx(k+1) + Du(k+1) + \zeta(k+1) && (2.41)\\
&= C(Ax(k) + Bu(k) + K\zeta(k)) + Du(k+1) + \zeta(k+1) && (2.42)\\
&= CAx(k) + \begin{bmatrix} CB & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} + \begin{bmatrix} CK & I \end{bmatrix} \begin{bmatrix} \zeta(k) \\ \zeta(k+1) \end{bmatrix} && (2.43)
\end{aligned}$$

by continuing this procedure for time instant $k+M-1$ can be written,

$$
\begin{aligned}
y(k+M-1) = CA^{M-1}x(k) + \begin{bmatrix} CA^{M-2}B & CA^{M-3}B & \dots & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+M-1) \end{bmatrix} \\
+ \begin{bmatrix} CA^{M-2}K & CA^{M-3}K & \dots & I \end{bmatrix} \begin{bmatrix} \zeta(k) \\ \zeta(k+1) \\ \vdots \\ \zeta(k+M-1) \end{bmatrix}
\end{aligned}
\tag{2.44}
$$

by compiling the output equations for time instants from k to $k+M-1$ following matrix equation is obtained,

$$
\begin{aligned}
\begin{bmatrix} y(k) \\ y(k+1) \\ \vdots \\ y(k+M-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{M-1} \end{bmatrix} x_k + \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{M-2}B & CA^{M-3}B & \dots & D \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+M-1) \end{bmatrix} \\
+ \begin{bmatrix} I & 0 & \dots & 0 & \\ CK & I & \dots & 0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{M-2}K & CA^{M-3}K & \dots & I & \end{bmatrix} \begin{bmatrix} \zeta(k) \\ \zeta(k+1) \\ \vdots \\ \zeta(k+M-1) \end{bmatrix}
\end{aligned}
\tag{2.45}
$$

By collecting the state variables in a vector as $[x(k) \;\; x(k+1) \;\; \dots \;\; x(k+M-1)]$ following block matrix output equation is obtained,

$$\begin{bmatrix} y(k) & y(k+1) & \dots & y(k+N-(M-1)) \\ y(k+1) & y(k+2) & \dots & y(k+N-(M-2)) \\ \vdots & \vdots & \vdots & \vdots \\ y(k+M-1) & y(M) & \dots & y(k+N) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{M-1} \end{bmatrix} \begin{bmatrix} x(k) & x(k+1) & \dots & x(k+M-1) \end{bmatrix}$$

$$+ \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{M-2}B & CA^{M-3}B & \dots & D \end{bmatrix} \begin{bmatrix} u(k) & u(k+1) & \dots & u(k+N-(M-1)) \\ u(k+1) & u(k+2) & \dots & u(k+N-(M-2)) \\ \vdots & \vdots & \vdots & \vdots \\ u(k+M-1) & u(M) & \dots & u(k+N) \end{bmatrix}$$

$$+ \begin{bmatrix} I & 0 & \dots & 0 \\ CK & I & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ CA^{M-2}K & CA^{M-3}K & \dots & I \end{bmatrix} \begin{bmatrix} \zeta(k) & \zeta(k+1) & \dots & \zeta(k+N-(M-1)) \\ \zeta(k+1) & \zeta(k+2) & \dots & \zeta(k+N-(M-2)) \\ \vdots & \vdots & \vdots & \vdots \\ \zeta(k+M-1) & \zeta(M) & \dots & \zeta(k+N) \end{bmatrix} \tag{2.46}$$

Therefore, the subspace I/O matrix equations are obtained from equations (2.39) and (2.46) as below,

$$Y_p = \Gamma_M X_p + H^d_M U_p + H^s_N Z_p \tag{2.47}$$

$$Y_f = \Gamma_M X_f + H^d_M U_f + H^s_N Z_f \tag{2.48}$$

$$X_f = A^M X_p + \Delta^d_M U_p + \Delta^s_N Z_p \tag{2.49}$$

where the superscripts d and s stand for the *deterministic* and *stochastic* part of the system, and subscripts p and f stand for the *past* and *future*. The past and future input-output (I/O) data and innovation block-Hankel matrices are defined as [20, 100],

$$U_p \triangleq \begin{bmatrix} u(1) & u(2) & \dots & u(N-2M+1) \\ u(2) & u(3) & \dots & u(N-2M+2) \\ \vdots & \vdots & \vdots & \vdots \\ u(M) & u(M+1) & \dots & u(N-M) \end{bmatrix}, U_f \triangleq \begin{bmatrix} u(M+1) & u(M+2) & \dots & u(N-M+1) \\ u(M+2) & u(M+3) & \dots & u(N-M+2) \\ \vdots & \vdots & \vdots & \vdots \\ u(2M) & u(2M+1) & \dots & u(N) \end{bmatrix} \tag{2.50}$$

$$Y_p \triangleq \begin{bmatrix} y(1) & y(2) & \dots & y(N-2M+1) \\ y(2) & y(3) & \dots & y(N-2M+2) \\ \vdots & \vdots & \vdots & \vdots \\ y(M) & y(M+1) & \dots & y(N-M) \end{bmatrix}, Y_f \triangleq \begin{bmatrix} y(M+1) & y(M+2) & \dots & y(N-M+1) \\ y(M+2) & y(M+3) & \dots & y(N-M+2) \\ \vdots & \vdots & \vdots & \vdots \\ y(2M) & y(2M+1) & \dots & y(N) \end{bmatrix} \tag{2.51}$$

$$Z_p \triangleq \begin{bmatrix} \zeta(1) & \zeta(2) & \dots & \zeta(N-2M+1) \\ \zeta(2) & \zeta(3) & \dots & \zeta(N-2M+2) \\ \vdots & \vdots & \vdots & \vdots \\ \zeta(M) & \zeta(M+1) & \dots & \zeta(N-M) \end{bmatrix}, Z_f \triangleq \begin{bmatrix} \zeta(M+1) & \zeta(M+2) & \dots & \zeta(N-M+1) \\ \zeta(M+2) & \zeta(M+3) & \dots & \zeta(N-M+2) \\ \vdots & \vdots & \vdots & \vdots \\ \zeta(2M) & \zeta(2M+1) & \dots & \zeta(N) \end{bmatrix} \tag{2.52}$$

where $U_p, U_f \in \mathbb{R}^{Mm \times N-2M+1}$, $Y_p, Y_f \in \mathbb{R}^{Ml \times N-2M+1}$ and $Z_p, Z_f \in \mathbb{R}^{Ml \times N-2M+1}$.

Note that, the data in the columns of the "future" block-Hankel matrices follows the data of the same column of the "past" block-Hankel matrices. In order to have an extra freedom in identification algorithm, the row dimension of the "past" and "future" block-Hankel matrices can be different. In SIM the data block-Hankel matrices should be very rectangular to minimize the noise sensitivity.

$$M > n, \quad N >> \max\{Mm, Ml\}, \quad N \to \infty$$

The "past" and "future" state matrices, X_p and X_f, are defined as,

$$X_p \triangleq \begin{bmatrix} x(1) & x(2) & x(3) & \dots & x(N-2M+1) \end{bmatrix} \in \mathbb{R}^{n \times N-2M+1} \tag{2.53}$$

$$X_f \triangleq \begin{bmatrix} x(M+1) & x(M+2) & x(M+3) & \dots & x(N-M+1) \end{bmatrix} \in \mathbb{R}^{n \times N-2M+1} \tag{2.54}$$

The extended $(M > n)$ observability matrix, Γ_M, and the reversed extended controllability matrices, Δ_M^d and Δ_M^s, are defined as,

$$\Gamma_M = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{M-1} \end{bmatrix} \in \mathbb{R}^{Ml\times n} \tag{2.55}$$

$$\Delta_M^d = \begin{bmatrix} A^{M-1}B & A^{M-2}B & \dots & AB & B \end{bmatrix} \in \mathbb{R}^{n\times Mm} \tag{2.56}$$

$$\Delta_M^s = \begin{bmatrix} A^{M-1}K & A^{M-2}K & \dots & AK & K \end{bmatrix} \in \mathbb{R}^{n\times Ml} \tag{2.57}$$

The lower triangular block-Toeplitz matrices, H_M^d and H_M^s, are also defined as below,

$$H_M^d = \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ CA^{M-2}B & CA^{M-3}B & CA^{M-4}B & \dots & D \end{bmatrix} \in \mathbb{R}^{Ml\times Mm} \tag{2.58}$$

$$H_M^s = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ CK & I & 0 & \dots & 0 \\ CAK & CK & I & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ CA^{M-2}K & CA^{M-3}K & CA^{M-4}K & \dots & I \end{bmatrix} \in \mathbb{R}^{Ml\times Ml} \tag{2.59}$$

Subspace Identification Methods (SIM)

Subspace identification algorithms basically consist of two main steps: First the system order, n, and the extended observability matrix of the system, Γ_M or the estimate of the state sequence X_f, are determined directly from the given I/O data. Then the system matrices (A, B, C, D, Q, R, S) are determined by using one of the following mentioned subspace identification algorithms up to similarity transformation. Two most popular subspace identification algorithms are: *Numerical Subspace State-Space System Identification* (N4SID), and *Multivariable Output Error State-Space* (MOESP).

- Numerical Subspace State-Space System Identification **(N4SID):** The algorithm constructs the projection of future output data onto past I/O data and estimates matrices via SVD and Least Squares, the algorithm was presented and expanded by Overschee and De Moor in 1990's [19, 20, 97]. This method mainly uses *oblique projection* concept and SVD for identification. The N4SID algorithm is summarized into four main steps,

Step1: Compute the oblique projection of future outputs onto the past I/O along the future inputs,

$$O_M = Y_f /_{U_f} W_p \tag{2.60}$$

where W_p is defined as the past I/O data matrix as below,

$$W_p = \begin{bmatrix} Y_p \\ U_p \end{bmatrix} \in \mathbb{R}^{M(m+l) \times N-2M+1} \tag{2.61}$$

From (2.48) the oblique projection can be obtained as,

$$O_M = \Gamma_M X_f /_{U_f} W_p + H_M^d U_f /_{U_f} W_p + H_N^s Z_f /_{U_f} W_p \tag{2.62}$$

From the definition of oblique projection we have $U_f /_{U_f} W_p = 0$, and when the number of samples N and number of past data row M are large enough we have,

$$\lim_{M,N \to \infty} X_f /_{U_f} W_p = \hat{X}_f \tag{2.63}$$

$$\lim_{N \to \infty} Z_f /_{U_f} W_p = 0 \tag{2.64}$$

Therefore, the oblique projection in (2.62) is simplified as,

$$O_M = \Gamma_M X_f \tag{2.65}$$

Step2: Compute the SVD of the oblique projection, then obtain system order $n = dim(\Sigma_1)$ and the estimate of the state vector X_f. The SVD of the oblique projection be given by,

$$O_M = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T \tag{2.66}$$

Since the oblique projection is $O_M = \Gamma_M X_f$, the extended observability matrix and the estimate of the state vector are derived from SVD as below,

$$\Gamma_M = U_1 \Sigma_1^{1/2} T \in \mathbb{R}^{Ml \times n} \tag{2.67}$$

$$X_f = T^{-1} \Sigma_1^{1/2} V_1^T \in \mathbb{R}^{n \times N-2M+1} \tag{2.68}$$

where $T \in \mathbb{R}^{n \times n}$ is an arbitrary non-singular matrix representing a similarity transformation.

Step3: In previous step the estimate of state vector is obtained from (2.68) as,

$$X_M = \left[\begin{array}{ccccc} x(M) & x(M+1) & \dots & x(N-M-1) & x(N-M) \end{array}\right] \tag{2.69}$$

then the following matrices with $N-2M$ columns are defined as,

$$\bar{X}_{M+1} = \left[\begin{array}{cccc} x(M+1) & x(M+2) & \dots & x(N-M) \end{array}\right] \tag{2.70}$$

$$\bar{X}_M = \left[\begin{array}{cccc} x(M) & x(M+1) & \dots & x(N-M-1) \end{array}\right] \tag{2.71}$$

$$\bar{U}_{M|M} = \left[\begin{array}{cccc} u(M) & u(M+1) & \dots & u(N-M-1) \end{array}\right] \tag{2.72}$$

$$\bar{Y}_{M|M} = \left[\begin{array}{cccc} y(M) & y(M+1) & \dots & y(N-M-1) \end{array}\right] \tag{2.73}$$

where $\bar{U}_{M|M}, \bar{Y}_{M|M}$ are block Hankel matrices with only one block row of inputs and outputs, respectively.

Step4: Compute the matrices A, B, C and D by solving the following regression equation by using the Least Squares technique,

$$\left[\begin{array}{c} \bar{X}_{M+1} \\ \bar{Y}_{M|M} \end{array}\right] = \left[\begin{array}{cc} A & B \\ C & D \end{array}\right] \left[\begin{array}{c} \bar{X}_M \\ \bar{U}_{M|M} \end{array}\right] \tag{2.74}$$

$$\left[\begin{array}{cc} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{array}\right] = \left(\left[\begin{array}{c} \bar{X}_{M+1} \\ \bar{Y}_{M|M} \end{array}\right] \left[\begin{array}{c} \bar{X}_M \\ \bar{U}_{M|M} \end{array}\right]^T \right) \left(\left[\begin{array}{c} \bar{X}_M \\ \bar{U}_{M|M} \end{array}\right] \left[\begin{array}{c} \bar{X}_M \\ \bar{U}_{M|M} \end{array}\right]^T \right)^{-1} \tag{2.75}$$

- *Multivariable Output Error State-Space* (MOESP): This method uses RQ decomposition of joint I/O data matrices and first calculates the extended observability matrix via SVD and obtain some of the state-space matrices, then uses Least Squares method to estimate other matrices. This method was introduced by Verhaegen and Dewilde in 1990's [19,101,102]. The MOESP algorithm is summarized into four main steps,

Step1: Suppose that the past I/O data U_p and Y_p were obtained. Compute the RQ decomposition of the past I/O data matrix as,

$$W_p = \begin{bmatrix} U_p \\ Y_p \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \tag{2.76}$$

where R_{11} and R_{22} are lower triangular and Q_1 and Q_2 are orthogonal. This RQ decomposition is computed by taking the transpose of the RQ decomposition of the W_p^T. The following equations are written as,

$$U_p = R_{11}Q_1^T \tag{2.77}$$

$$Y_p = R_{21}Q_1^T + R_{22}Q_2^T \tag{2.78}$$

Suppose that the following conditions are satisfied, (i) the system is reachable or the state vector is sufficiently excited, (ii) the system is persistently excited by the input sequence and (iii) the I/O data are obtained from an open-loop experiment. Under these assumption R_{11} is nonsingular, so that (2.78) is written as follows,

$$Y_p = R_{21}R_{11}^{-1}U_p + R_{22}Q_2^T \tag{2.79}$$

since Q_1^T, Q_2^T are orthogonal, the first term in the right-hand side of the equation (2.78) is orthogonal to the second term. The orthogonal projection of the row space of Y_p onto the row space of U_p and its complement $U_p^{\perp}$ is given by these two equations,

$$Y_p/U_p = R_{21}Q_1^T = R_{21}R_{11}^{-1}U_p \tag{2.80}$$

$$Y_p/U_p^{\perp} = R_{22}Q_2^T \tag{2.81}$$

also, it follows from equations (2.47) and (2.76) that,

$$\Gamma_M X_p + H_M^d R_{11}Q_1^T = R_{21}Q_1^T + R_{22}Q_2^T \tag{2.82}$$

post-multiplying (2.82) by Q_2 yields,

$$\Gamma_M X_p Q_2 = R_{22} \tag{2.83}$$

now, the image of the extended observability matrix Γ_M and the dimension n obtain from SVD of R_{22}.

Step2: Compute the SVD of R_{22} to obtain the system order, n, and the extended observability matrix, Γ_M. The SVD of R_{22} be given by,

$$R_{22} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T \tag{2.84}$$

then $\Gamma_M X_p Q_2 = U_1 \Sigma_1 V_1^T$ and the extended observability matrix is defined as,

$$\Gamma_M = U_1 \Sigma_1^{1/2} \tag{2.85}$$

and the order is $n = dim(\Sigma_1)$.

Step3: Obtain the matrices C and A by using the extended observability matrix. The matrix C is given by,

$$C = \Gamma_M(1 : l, 1 : n) \tag{2.86}$$

and A is obtained by solving this equation,

$$\Gamma_M(1 : l(M-1), 1 : n)A = \Gamma_M(l+1 : Ml, 1 : n) \tag{2.87}$$

Step4: Estimate the matrices B and D by the Least Squares method. Since $U_2^T R_{22} = 0$ and $U_2^T \Gamma_M = 0$, pre-multiplying equation (2.82) by U_2^T, then post-multiplying it by Q_1 yields,

$$U_2^T H_M^d R_{11} Q_1^T = U_2^T R_{21} Q_1^T \longrightarrow U_2^T H_M^d R_{11} = U_2^T R_{21} \tag{2.88}$$

Now, there is a linear equation with respect to B and D, it can be solved by Least Squares method,

$$U_2^T \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ CA^{M-2}B & CA^{M-3}B & CA^{M-4}B & \dots & D \end{bmatrix} = U_2^T R_{21} R_{11}^{-1} \tag{2.89}$$

by Least Squares solution of this overdetermined linear matrix equation B and D can be estimated.

Figure 2.2 shows the overview of N4SID and MOESP methods. Through the N4SID and MOESP algorithms the state sequence X_f and the extended observability matrix Γ_M are determined. Then the system matrices A, B, C and D are extracted using the algorithms.

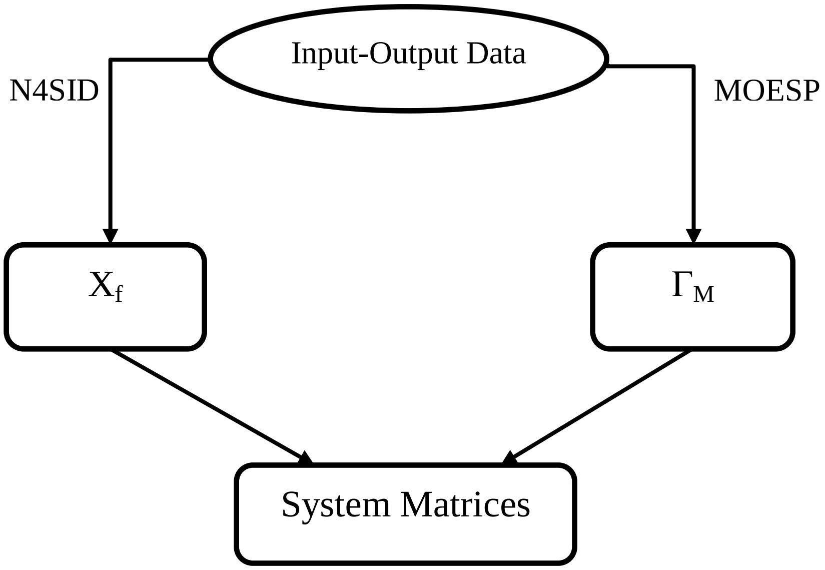

Figure 2.2: Comparing the N4SID and MOESP algorithms

2.2 Predictive Control

In last three decades Predictive Controllers became most widely used advanced controllers in the process control in industry [7, 10, 103]. In general, predictive control is a control strategy which obtains the control signal by using a predictive model of the process in a cost function minimization. The procedure is done over a fixed prediction horizon in the presence of constraints. There are two different approaches for predictive control. First is the *model-based* approach, which is called Model-based Predictive Control (MBPC) or briefly Model Predictive Control (MPC) [7,12,15,104]. Second the *model-free* approach, which is called Data-driven Subspace-based Predictive Control (SBPC) or briefly Subspace Predictive Control (SPC) [18,21,100].

Process model is the basic requirement of the design in model-based approach. Then the predictor matrices are obtained from the process model, and the controller is design by using the predictor matrices. Therefore, in MPC the closed loop performance of the system heavily depends on the accuracy of the process model, which is utilized to design the predictor. Hence, modeling is considered as the most challenging and time consuming work in MPC design.

On the other hand, in model-free approach the predictor matrices are determined directly from I/O data by using the data block-Hankel matrices that are defined in SIM with no need to obtain the parametric model of the process and the identification step. Hence, model-free approach is also called *data-driven* approach. In model-free approach the predictor is derived from the subspace I/O matrix equations of SIM in (2.47), (2.48) and (2.49), which is called *Subspace Predictor*. Therefore, the model-free approach is obtained based on combination of the Subspace Predictor and Predictive Control which is called SPC. Figure 2.3 illustrate the difference between model-based approach and data-driven approach for predictive control. Main advantage of SPC over MPC is its capability to on-line adapt the SPC gains by collecting new I/O data and updating the Subspace Predictor matrices in data-driven manner. This feature makes SPC much more appropriate than MPC to control time-varying and nonlinear systems.

2.2.1 Model-based Approach: Model Predictive Control (MPC)

Note that while the concentration of this dissertation is on SPC (Section 2.2.2), the MPC and SPC methods share the same cost function and SPC has been introduced after MPC. This section concentrates on formulation of MPC and the provided cost function, control law and constraint of this section are used in SPC approaches.

There are different MPC techniques in the literature, and some of the most popular MPC algorithms are Dynamic Matrix Control (DMC) [13], Quadratic Dynamic Matrix Control (QDMC) [14], and Generalized Predictive Control (GPC) [15]. In general, all of the MPC algorithms have three basic elements:

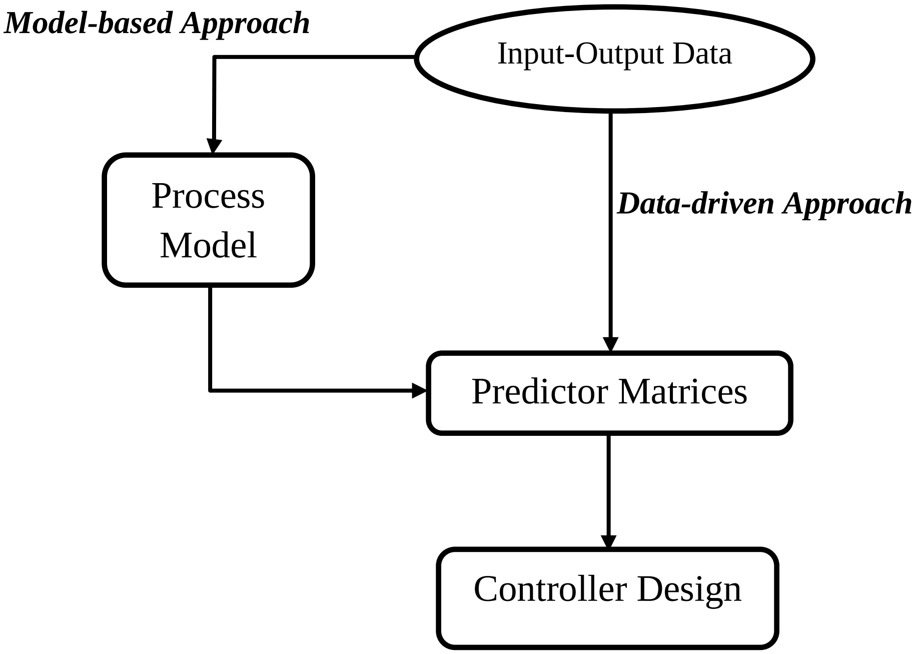

Figure 2.3: Comparing model-based approach and data-driven approach in MPC

Prediction Models

MPC is a *model-based* optimization technique, however, the models are not directly used to design the controller. Instead, they are used to design the predictors, and then the control strategy is obtained according to the predictor. In MPC the predictor is multi-step ahead not just one-step ahead, therefore, the system output follows a trajectory of the reference input. Moreover, in MPC future outputs are predicted by considering the future inputs, not only by past inputs and outputs. Some of the typical models used in MPC are [98]:

- *Step response model,* which is used in DMC algorithm [98].

$$y(k) = \sum_{i=1}^{\infty} g_i \Delta u(k-i) \tag{2.90}$$

where g_is are the sampled unit step values, $\Delta = 1 - q^{-1}$ with q^{-1} as the back shift operator, and $\Delta u(k) = u(k) - u(k-1)$. In step response modeling there is no need to a priori information about the system, and complex dynamics such as non-minimum phase or time-delays can be described easily. However, it requires large number of parameters, and it is limited to stable systems without integrators.

The step response model is used in DMC algorithm to obtain the prediction model. Assume that the system is stable and N_s is the settling time in terms of samples. The step response model in (2.90) can be written as,

$$y(k) = \sum_{i=1}^{N_s} g_i \Delta u(k-i) + \sum_{i=N_s+1}^{\infty} g_i \Delta u(k-i) \tag{2.91}$$

using the step response model in (2.91) the prediction of the output at the instant k until the instant $k + N_p$ under the effect of N_c control actions can be obtain as follows,

$$\hat{y}_f = y_f^* + G\Delta u_f \tag{2.92}$$

where the index f represents *future* data, $\hat{y}_f$ is the predicted output, y_f^* represents free response of the system, Δu_f is variation of the future control signal and G is called the *dynamic matrix*, which contains the system dynamics,

$$\hat{y}_f = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N_p|k) \end{bmatrix}, \quad y_f^* = \begin{bmatrix} y^*(k+1) \\ y^*(k+2) \\ \vdots \\ y^*(k+N_p) \end{bmatrix}, \quad \Delta u_f = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix} \tag{2.93}$$

$$G_{N_p} = \begin{bmatrix} g_1 & 0 & \dots & 0 \\ g_2 & g_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N_c} & g_{N_c-1} & \cdots & g_1 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N_p} & g_{N_p-1} & \cdots & g_{N_p-N_c+1} \end{bmatrix} \tag{2.94}$$

Free response of the system is defined as the response that would be expected if no future actions are taken. Thus $y^*(k+N_p)$ is calculated as,

$$y^*(k+N_p) = \sum_{i=N_p+1}^{N_s} g_i \Delta u(k+N_p-i) + \sum_{i=N_s+1}^{\infty} g_i \Delta u(k+N_p-i) \tag{2.95}$$

- *Transfer function models,* which are used in GPC design [98].

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k) + \frac{C(q^{-1})}{A(q^{-1})}w(k) \tag{2.96}$$

These models needs fewer parameters, and applicable for unstable systems. However, structure selection (ARX, ARMAX, BJ, OE) and order of the polynomials is necessary for identification. Transfer function models lead to solve the Diophantine equation in optimization procedure. Also, parametric models with limited number of parameters may not be suitable for some processes and MIMO systems.

GPC design starts with identifying an ARIMAX (integrated moving-average) model for the system. ARIMAX model is expressed as,

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k-1) + \frac{C(q^{-1})}{A(q^{-1})}\frac{w(k)}{\Delta} \tag{2.97}$$

Optimal prediction is obtained by recursion of following Diophantine equations,

$$\begin{aligned} \frac{C(q^{-1})}{A(q^{-1})\Delta} &= G_{di}(q^{-1}) + q^{-1}\frac{F(q^{-1})}{A(q^{-1})\Delta} && (2.98)\\ G_{di}(q^{-1})B(q^{-1}) &= G_i(q^{-1})C(q^{-1}) + q^{-1}\Gamma_i(q^{-1}) && (2.99) \end{aligned}$$

The N_p step ahead prediction equation is written as,

$$\hat{y}(k+N_p|k) = y^*(k+N_p) + G_{N_p}\Delta u(k+N_p-1) + G_{d_{N_p}}d(k+N_p) \tag{2.100}$$

where, the free response of the system, $y^*(k+N_p)$, is calculated as,

$$y^*(k+N_p) = \frac{F_{N_p}(q^{-1})}{C(q^{-1})}y(k) + \frac{T_{N_p}(q^{-1})}{C(q^{-1})}\Delta u(k) \tag{2.101}$$

by ignoring the future disturbance term $G_{d_i}d(k+i)$, the multi-step predictor equation is expressed as,

$$\hat{y}_f = y_f^* + G\Delta u_f \tag{2.102}$$

where the index f represents *future* data, $\hat{y}_f$ is the predicted output, y_f^* represents free response of the system, Δu_f is variation of the future control signal,

$$\hat{y}_f = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N_p|k) \end{bmatrix} , \ y_f^* = \begin{bmatrix} y^*(k+1) \\ y^*(k+2) \\ \vdots \\ y^*(k+N_p) \end{bmatrix} , \ \Delta u_f = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix} \tag{2.103}$$

and G is the *dynamic matrix*, which contains the step response coefficients of $\frac{B(q^{-1})}{A(q^{-1})}$ or the impulse response coefficients of $\frac{B(q^{-1})}{A(q^{-1})\Delta}$.

- *State-space models,* which became more popular recently due to applicability for stable, unstable and MIMO systems [98],

$$x(k+1) = Ax(k) + Bu(k) \tag{2.104}$$

$$y(k) = Cx(k) + Du(k) \tag{2.105}$$

however, process modeling with limited number of parameters is one of the disadvantages of these modeling method. Also, using an observer may complicate the calculations if all of the states are not reachable.

Assume state-space model of a MIMO system by considering the system noise and disturbance as below [12],

$$\bar{x}(k+1) = \bar{A}x(k) + \bar{B}u(k) + \bar{B}_d d(k) \tag{2.106}$$

$$y(k) = \bar{C}\bar{x}(k) \tag{2.107}$$

where $\bar{A} \in \mathbf{R}^{n_1 \times n_1}, \bar{B} \in \mathbf{R}^{n_1 \times m}$ and $\bar{C} \in \mathbf{R}^{l \times n_1}$ are the state-space matrices for system model, $\bar{x}(k) \in \mathbf{R}^{n_1 \times 1}$ is the state variable vector, $d(k)$ is the input disturbance that is a sequence of integrated white noise. Note that from the principle of predictive control the $u(k)$ cannot affect the $y(k)$ at the same time. Thus, $\bar{D} = 0$ in the system model.

State-space based MPC design starts with deriving the Integrator Embedded State-space (IESS) model as,

$$\underbrace{\begin{bmatrix} \Delta\bar{x}(k+1) \\ y(k+1) \end{bmatrix}}_{x(k+1)} = \underbrace{\begin{bmatrix} \bar{A} & \mathbf{0}^T_{n_1\times 1} \\ \bar{C}\bar{A} & I_{l\times l} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \Delta\bar{x}(k) \\ y(k) \end{bmatrix}}_{x(k)} + \underbrace{\begin{bmatrix} \bar{B} \\ \bar{C}\bar{B} \end{bmatrix}}_{B} \Delta u(k) + \underbrace{\begin{bmatrix} \bar{B}_d \\ \bar{C}\bar{B}_d \end{bmatrix}}_{B_w} w(k) \tag{2.108}$$

$$y(k) = \underbrace{\begin{bmatrix} \mathbf{0}_{n_1\times 1} & I_{l\times l} \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} \Delta\bar{x}(k) \\ y(k) \end{bmatrix}}_{x(k)} \tag{2.109}$$

Therefore the IESS model is obtained as,

$$x(k+1) = Ax(k) + B\Delta u(k) + B_w w(k) \tag{2.110}$$

$$y(k) = Cx(k) \tag{2.111}$$

The triplet (A, B, C) is called the *augmented model* which is used in MPC design. The dimensionality of the augmented state-space system is $n = n_1 + l$.

Assuming that the state variable vector $x(k)$ is available at the sampling time k, the N_p step ahead state variables are predicted as,

$$\begin{aligned} \hat{x}(k+N_p|k) = A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + \cdots + A^{N_p-N_c}B\Delta u(k+N_c-1) \\ +A^{N_p-1}B_w w(k) + A^{N_p-2}B_w w(k+1|k) + \cdots + B_w w(k+N_p-1|k) \end{aligned} \tag{2.112}$$

From the predicted state variables, the N_p step a head predicted output variables are,

$$\begin{aligned} \hat{y}(k+N_p|k) = CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + \cdots + CA^{N_p-N_c}B\Delta u(k+N_c-1) \\ +CA^{N_p-1}B_w w(k) + \cdots + CA^{N_p-N_c}B_w w(k+N_c-1) \end{aligned} \tag{2.113}$$

Note that, for zero-mean white noise sequence $w(k)$, the predicted value of $w(k+i-1|k)$ at future sample i is assumed to be zero. Therefore, the predictor equation is defined as,

$$\hat{y}_f = \Upsilon x(k) + \Phi\Delta u_f \tag{2.114}$$

where the index f represents *future* data, $\hat{y}_f$ is the predicted output, Δu_f is the future control signal variations,

$$\hat{y}_f = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N_p|k) \end{bmatrix}, \quad \Delta u_f = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix} \tag{2.115}$$

and Υ and Φ are defined as,

$$\Upsilon = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \quad \Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix} \tag{2.116}$$

Cost Function and Tuning Parameters

There are different cost functions for different MPC techniques, but all of them wants to minimize the output deviation from the reference input by optimizing the control signal or its variation. A typical form of such a cost function in MPC is [12,68,98],

$$J(k) = \sum_{i=N_d}^{N_p} (\hat{y}(k+i|k) - r(k+i))^T Q (\hat{y}(k+i|k) - r(k+i)) + \sum_{i=1}^{N_c} \Delta u^T(k+i) R \Delta u(k+i) \tag{2.117}$$

where $r(k)$ is the reference signal at the current sample time k, N_p is the *Prediction Horizon* and N_c is the *Control Horizon.* N_d is usually chosen as one or the system time-delay in terms of samples. $Q \in \mathbb{R}^{N_pl \times N_pl}$ and $R \in \mathbb{R}^{N_cm \times N_cm}$ are the positive semi-definite and positive definite *Weighting Matrices* for penalizing the tracking error and the incremental control signal, respectively. The weighting matrices and horizons are considered as *Tuning Parameters* to ensure the stability and enhance performance of the closed-loop system in MPC design. The cost function in (2.117) can be written in vector form as,

$$J = (\hat{y}_f - r_f)^T Q (\hat{y}_f - r_f) + \Delta u_f^T R \Delta u_f \tag{2.118}$$

where r_f is a vector containing the future reference inputs as,

$$r_f = \begin{bmatrix} r(k+1) \\ r(k+2) \\ \vdots \\ r(k+N_p) \end{bmatrix} \tag{2.119}$$

In MPC algorithm, at the current sample time k, first the appropriate future input trajectory is calculated based on the past I/O data and the internal model of the system over the control horizon N_c, then the best future output is predicted using the *model-based* designed predictor over a fixed prediction horizon N_p. The prediction horizon needs to be selected large enough to ensure the stability and tracking purposes. This can be achieved by applying following *terminal equality constraint* [43, 44],

$$e(k+N_p+i|k) = 0, \qquad \forall i \geq 0 \tag{2.120}$$

where $e(k+i) = r(k+i) - \hat{y}(k+i|k)$ is the tracking error. The Figure 2.4 shows a typical MPC strategy at the time instant k. Note that, the input trajectory remains constant after the control

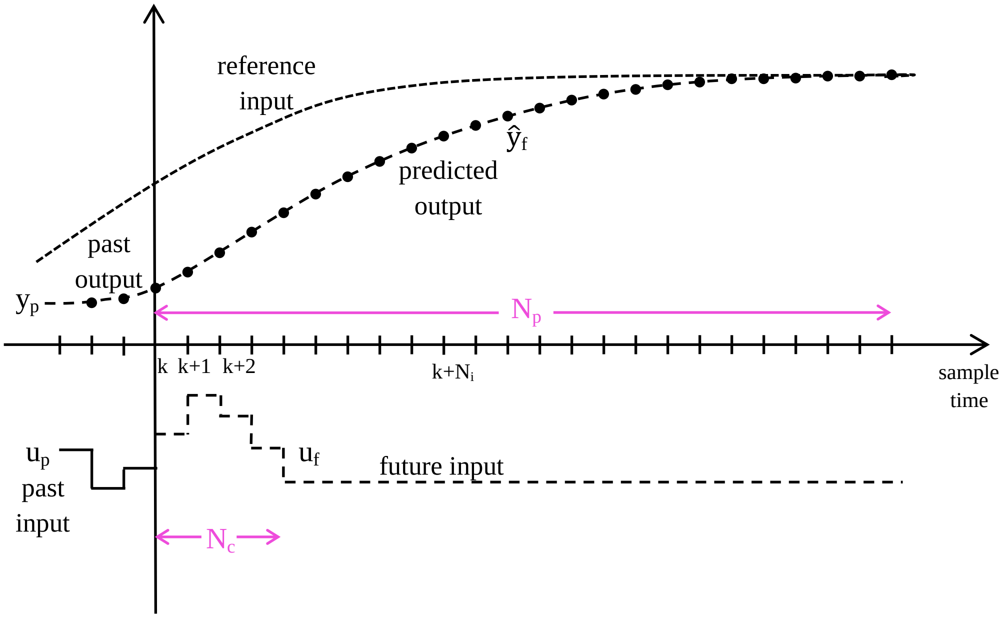

Figure 2.4: A typical MPC strategy

horizon N_c, which is applied by adding following *equality constraint* to the cost function [43,44],

$$\Delta u(k + N_c + i) = 0, \qquad \forall i \geq 0 \tag{2.121}$$

In MPC algorithms after calculating the future control trajectory, only its first element is applied to the system. At the next time step $k + 1$, the whole procedure is repeated by one time step shifting forward of the both horizons.

Control Law and Constraints

In MPC the control law, $u(k) = u(k-1) + \Delta u(k)$ is calculated via minimizing the cost function and solving a typical Least Squares problem as [12,43,44,98,105],

$$\begin{aligned} \min_{\Delta u_f} \quad & J = (\hat{y}_f - r_f)^T Q(\hat{y}_f - r_f) + \Delta u_f^T R \Delta u_f \\ \text{subject to} \quad & \hat{y}_f \quad \text{from (2.92) or (2.102) or (2.114)} \\ & e(k + N_p + i|k) = 0, \qquad \forall i \geq 0 \\ & \Delta u(k + N_c + i) = 0, \qquad \forall i \geq 0 \end{aligned} \tag{2.122}$$

where $\hat{y}_f$ could be any of the predictor equations of DMC in (2.92), GPC in (2.102) or state-space based MPC in (2.114). The unconstrained MPC solution is found from,

$$\frac{\partial J}{\partial \Delta u_f} = 0 \tag{2.123}$$

for DMC and GPC:

$$\Delta u_f = (G^T Q^T G + R)^{-1} G^T Q^T (r_f - y_f^*) \tag{2.124}$$

and for state-space MPC:

$$\Delta u_f = (\Phi^T Q^T \Phi + R)^{-1} \Phi^T Q^T (r_f - \Upsilon x(k)) \tag{2.125}$$

However, in the presence of active *inequality constraints* on $u(k)$, $\Delta u(k)$, $y(k)$ or $\Delta y(k)$ using the numerical optimization is necessary. Capability to handle the inequality constraints is one of the main advantages of MPC algorithms. The inequality constraints are specified in the form of,

$$\begin{array}{ccccc} u_{min} & \leq & u & \leq & u_{max} \\ \Delta u_{min} & \leq & \Delta u & \leq & \Delta u_{max} \\ y_{min} & \leq & y & \leq & y_{max} \\ \Delta y_{min} & \leq & \Delta y & \leq & \Delta y_{max} \end{array} \tag{2.126}$$

Therefore the constrained MPC problem is summarized as follows [12,43,44,98],

$$\begin{aligned} \min_{\Delta u_f} \quad & J = (\hat{y}_f - r_f)^T Q (\hat{y}_f - r_f) + \Delta u_f^T R \Delta u_f \\ \text{subject to} \quad & \hat{y}_f \quad \text{from (2.92) or (2.102) or (2.114)} \\ & e(k + N_p + i|k) = 0, \qquad \forall i \geq 0 \\ & \Delta u(k + N_c + i) = 0, \qquad \forall i \geq 0 \\ & v_{min} \leq v \leq v_{max} \end{aligned} \tag{2.127}$$

where v is any of the constrained variables in (2.126).

2.2.2 Data-Driven Approach: Subspace Predictive Control (SPC)

The data-driven SPC approach is derived via the combination of Subspace predictor and Predictive control technique. SPC has some features, which has made it as one of the most popular control strategies in industry over the past decade [21,66,68]. These features are:

1- No need of prior knowledge on the order and model structure of the system,

2- On-line adapting the controller in data-driven manner,

3- Same cost function and same tuning parameters as GPC,

4- No need to solve the Diophantine equation,

5- Based on the reliable linear algebra tools such as SVD and QR decomposition,

6- No need for iterative algorithm to obtain predictor matrices,

7- Easily applicable to MIMO systems.

Subspace Predictor

Consider the state-space innovation form of discrete LTI system from (2.26) and (2.27). Suppose that measurements of the I/O data, $u(k)$ and $y(k)$ are available for $k \in \{1, 2, \ldots, N\}$, and the data is persistently exciting. The I/O data and innovation block-Hankel matrices are constructed as equations (2.50), (2.51) and (2.52). Note that, to minimize the unwanted effect of noise on the identification of the system, the number of columns in these data block-Hankel matrices should be much larger than the number of rows. Moreover, the input $u(k)$ is persistently exciting and it is uncorrelated with $\zeta(k)$, and the number of measurements are large enough ($N \to \infty$) [18, 21, 68].

Consider the subspace future output matrix equation in (2.48), by substituting equations (2.47) and (2.49) into (2.48) the future output matrix equation is obtained as below,

$$Y_f = \Gamma_M A^M \Gamma_M^{\dagger} Y_p + \Gamma_M (\Delta_M^d - A^M \Gamma_M^{\dagger} H_M^d) U_p + H_M^d U_f + \Gamma_M (\Delta_M^s - A^M \Gamma_M^{\dagger} H_M^s) Z_p + H_M^s Z_f \quad (2.128)$$

where superscript † represent the Moore-Penrose pseudo-inverse. Since Z_f is stationary white noise and for a large enough set of measurements, the optimal prediction of Y_f from equation (2.128) is written as a linear combination of past I/O data and future input of the system [18, 21]. Therefore the *Subspace Predictor* equation is defined as,

$$\hat{Y}_f = L_w W_p + L_u U_f \quad (2.129)$$

where $L_w \in \mathbb{R}^{Ml \times M(l+m)}$ and $L_u \in \mathbb{R}^{Ml \times Mm}$ are the *subspace linear predictor coefficient matrices* that corresponds to the past I/O data and the future input data, respectively. W_p is the past I/O data matrix which is defined in (2.61).

The subspace predictor matrices L_w and L_u are calculated from the data block-Hankel matrices, by solving the following Least Squares problem,

$$\min_{L_w, L_u} \|Y_f - [L_w \quad L_u] \begin{bmatrix} W_p \\ U_f \end{bmatrix} \|_F^2 \quad (2.130)$$

The Least Squares problem can be solved numerically by making the following RQ-decomposition,

$$\begin{bmatrix} W_p \\ U_f \\ Y_f \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{bmatrix} = R^T Q^T \quad (2.131)$$

where R^T is a lower triangular matrix and Q is an orthogonal matrix. Therefore, $\hat{Y}_f$ can be written as below,

$$\hat{Y}_f = [R_{31} \quad R_{32}] \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix}^{\dagger} \begin{bmatrix} W_p \\ U_f \end{bmatrix} \tag{2.132}$$

where superscript † represent the Moore-Penrose pseudo-inverse. From equations (2.129) and (2.132) the matrices L_w and L_u are obtained as below,

$$L = [L_w \quad L_u] = [R_{31} \quad R_{32}] \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix}^{\dagger} \tag{2.133}$$

where, L_w and L_u can be represented as [18,21],

$$L_w = L(: ,\ 1 : M(m+l)) \tag{2.134}$$

$$L_u = L(: ,\ M(m+l)+1 : end) \tag{2.135}$$

assuming that M is large enough ($M \to \infty$) following results are obtained from subspace identification [18,21],

$$L_w W_p = \Gamma_M X_f \tag{2.136}$$

$$L_u = H_M^d \tag{2.137}$$

Therefore, the linear subspace predictor can describe the system behavior by constructing the subspace matrices directly from I/O data without need to identify the system model.

Unconstrained SPC

The developed subspace predictor in (2.129) can be combined with MPC cost function in (2.118) to obtain the data-driven approach SPC [18,21]. The SPC problem is realized by minimization of a cost function under certain constraints on the control signal and the system output.

In implementation of SPC, only the leftmost column of $\hat{Y}_f$ is considered to predict the output. Selecting the Prediction Horizon, N_p, and the Control Horizon, N_c, truncates the subspace predictor coefficient matrices. Therefore, (2.129) is rewritten as,

$$\hat{y}_f = \tilde{L}_w w_p + \tilde{L}_u u_f \tag{2.138}$$

where $\tilde{L}_w$ and $\tilde{L}_u$ are defined as follows,

$$\tilde{L}_w = L_w(1:N_pl,:) \tag{2.139}$$

$$\tilde{L}_u = L_u(1:N_pl,1:N_cm) \tag{2.140}$$

Here, $\hat{y}_f$, u_f and w_p are defined as the leftmost column of matrices $\hat{Y}_f$, U_f and W_p as,

$$\hat{y}_f = \begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N_p|k) \end{bmatrix}, \quad u_f = \begin{bmatrix} u(k+1) \\ u(k+2) \\ \vdots \\ u(k+N_c) \end{bmatrix}, \quad w_p = \begin{bmatrix} y_p \\ u_p \end{bmatrix} = \begin{bmatrix} y(k-M+1) \\ \vdots \\ y(k-1) \\ y(k) \\ ---- \\ u(k-M+1) \\ \vdots \\ u(k-1) \\ u(k) \end{bmatrix} \tag{2.141}$$

In the case of no active constraints, a typical form of cost function can be considered as (2.117) or (2.118). Recall the cost function from (2.118), the SPC control law which is applied to the system will be obtained by replacing the (2.138) in the cost function J in (2.118), and minimizing the cost function [18, 21]. Therefore, the predicted output, in (2.138) needs to be written in terms of incremental input signal as below [21],

$$\hat{y}_f = F_1y(k) + F_2\tilde{L}_w\Delta w_p + F_2\tilde{L}_u\Delta u_f \tag{2.142}$$

where incremental input signal Δu_f, incremental I/O data Δw_p, F_1 and F_2 are defined as,

$$\Delta u_f = \begin{bmatrix} \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c) \end{bmatrix}, \quad \Delta w_p = \begin{bmatrix} \Delta y_p \\ \Delta u_p \end{bmatrix} = \begin{bmatrix} \Delta y(k-M+1) \\ \vdots \\ \Delta y(k-1) \\ \Delta y(k) \\ ---- \\ \Delta u(k-M+1) \\ \vdots \\ \Delta u(k-1) \\ \Delta u(k) \end{bmatrix} \tag{2.143}$$

$$F_1 = \begin{bmatrix} I_l \\ I_l \\ \vdots \\ I_l \end{bmatrix}, \quad F_2 = \begin{bmatrix} I_l & 0 & \dots & 0 \\ I_l & I_l & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_l & I_l & \dots & I_l \end{bmatrix} \tag{2.144}$$

The unconstrained SPC problem is defined as follows,

$$\begin{aligned} \min_{\Delta u_f} \quad & J = (\hat{y}_f - r_f)^T Q (\hat{y}_f - r_f) + \Delta u_f^T R \Delta u_f \\ \text{subject to} \quad & \hat{y}_f = F_1 y(k) + F_2 \tilde{L}_w \Delta w_p + F_2 \tilde{L}_u \Delta u_f \end{aligned} \tag{2.145}$$

by replacing the (2.142) in the cost function J in (2.118), and minimizing the cost function, the incremental control law will be obtained as,

$$\frac{\partial J}{\partial \Delta u_f} = 0$$

$$\Delta u_f = -(R + (F_2\tilde{L}_u)^T Q (F_2\tilde{L}_u))^{-1} (F_2\tilde{L}_u)^T Q ((F_2\tilde{L}_w)\Delta w_p + F_1 y(k) - F_1 r(k+1)) \tag{2.146}$$

Remark 2.2.1. *Different control signals can be obtained from (2.146) by tuning the SPC parameters Q, R, N_p and N_c. Therefore, appropriate choice of these tuning parameters has a significant impact on the performance and stability of the closed-loop system.*

The incremental control law in (2.146) can be summarized as below;

$$\Delta u_f \quad = \quad -\tilde{K}_e(y(k) - r(k+1)) - \tilde{K}_{\Delta w_p}\Delta w_p \tag{2.147}$$

where *SPC gain matrices* $\tilde{K}_e$ and $\tilde{K}_{\Delta w_p}$ are defined as,

$$\tilde{K}_e \quad = \quad (R + (F_2\tilde{L}_u)^T Q(F_2\tilde{L}_u))^{-1}(F_2\tilde{L}_u)^T QF_1 \tag{2.148}$$

$$\tilde{K}_{\Delta w_p} \quad = \quad (R + (F_2\tilde{L}_u)^T Q(F_2\tilde{L}_u))^{-1}(F_2\tilde{L}_u)^T Q(F_2\tilde{L}_w) \tag{2.149}$$

Since, at each time instant only the first element of Δu_f is considered to calculate the next control input, then by truncating the first m rows in (2.146), the control law at time instant $k+1$ is obtained as,

$$\Delta u(k+1) \quad = \quad -\tilde{k}_e(y(k) - r(k+1)) - \tilde{k}_{\Delta w_p}\Delta w_p \tag{2.150}$$

$$u(k+1) \quad = \quad u(k) + \Delta u(k+1) \tag{2.151}$$

where the controller gains $\tilde{k}_e$ and $\tilde{k}_{\Delta w_p}$ are defined as the first m rows of the gain matrices $\tilde{K}_e$ and $\tilde{K}_{\Delta w_p}$, respectively. By using this control action zero steady-state tracking error is achievable, even for systems with no integrator. Note that above-mentioned SPC control law in (2.147) is obtained directly from subspace matrices, with no need to any system modeling and identification. Figure 2.5 shows a typical block diagram of data-driven SPC approach.

Constrained SPC

One of the main features of MPC and SPC is their ability to include constraints in the control algorithm. In real processes, there are constraints that are raised due to physical limitation of the system and safety concerns. Generally, in MIMO systems, the constrained SPC problem is considered as a Quadratic Programming (QP) problem and solved by a QP function solver to

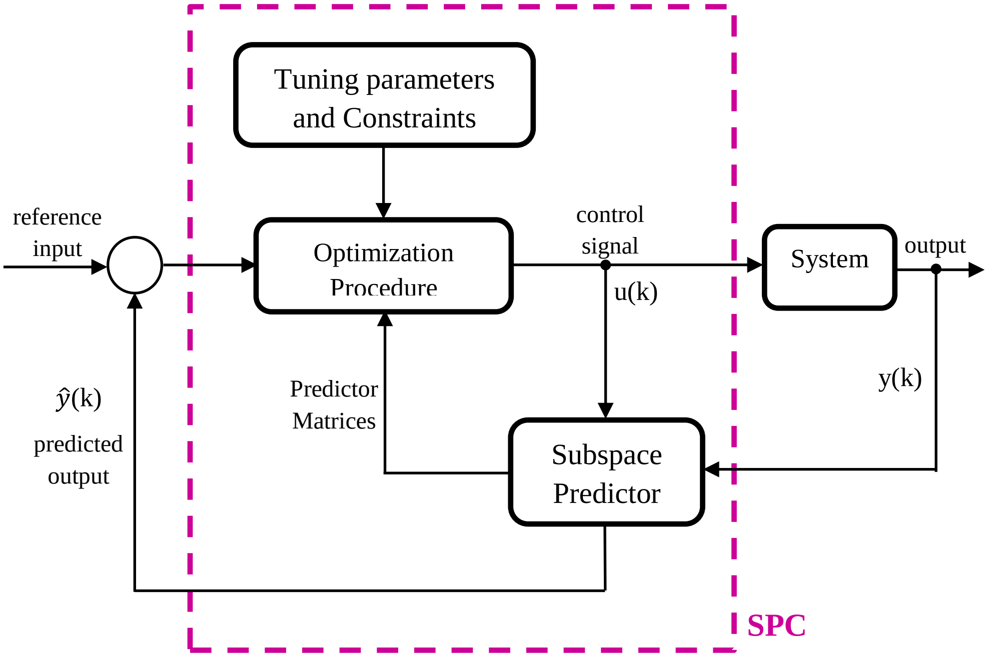

Figure 2.5: A Typical Subspace Predictive Control System

obtain the most optimum control law [21, 68]. The constrained SPC problem is defined as,

$$\begin{aligned} \min_{\Delta u_f} \quad & J = (\hat{y}_f - r_f)^T Q(\hat{y}_f - r_f) + \Delta u_f^T R \Delta u_f \\ \text{subject to} \quad & \hat{y}_f = F_1 y(k) + F_2 \tilde{L}_w \Delta w_p + F_2 \tilde{L}_u \Delta u_f \\ & e(k + N_p + i|k) = 0, \qquad \forall i \geq 0 \\ & \Delta u(k + N_c + i) = 0, \qquad \forall i \geq 0 \\ & v_{min} \leq v \leq v_{max} \end{aligned} \tag{2.152}$$

where v is any of the constrained variables, which are defined in (2.126).

To formulate the constrained SPC problem in QP form the SPC cost function needs to be revised as follows: The SPC cost function in (2.118) can be rewritten by using the predicted output in (2.142) and reference input in (2.119) as below,

$$\begin{aligned} J = {} & (F_2 \tilde{L}_w \Delta w_p + F_2 \tilde{L}_u \Delta u_f + F_1(y(k) - r(k+1)))^T Q(F_2 \tilde{L}_w \Delta w_p + F_2 \tilde{L}_u \Delta u_f + F_1(y(k) - r(k+1))) \\ & + \Delta u_f^T R \Delta u_f \end{aligned} \tag{2.153}$$

by expanding (2.153) and removing the terms that are not related to Δu_f the QP at every instant will be obtained as [21, 68],

$$\min_{\Delta u_f} \quad J \quad = \quad \Delta u_f^T H \Delta u_f + P^T \Delta u_f \tag{2.154}$$

where H and P are defined as,

$$\begin{aligned} H &= (F_2 \tilde{L}_u)^T Q (F_2 \tilde{L}_u) + R && (2.155) \\ P &= 2(F_2 \tilde{L}_u)^T Q (F_2 L_w \Delta w_p + F_1(y(k) - r(k+1))) && (2.156) \end{aligned}$$

The constraints for SPC are rewritten as,

$$\begin{array}{ccc} F_4 \Delta u_f & \leq & F_3 u_{max} - F_3 u(k) \\ -F_4 \Delta u_f & \leq & -F_3 u_{min} + F_3 u(k) \\ \Delta u_f & \leq & F_3 \Delta u_{max} \\ -\Delta u_f & \leq & -F_3 \Delta u_{min} \\ F_2 \tilde{L}_u \Delta u_f & \leq & F_1 y_{max} - F_1 y(k) - F_2 \tilde{L}_w \Delta w_p \\ -F_2 \tilde{L}_u \Delta u_f & \leq & -F_1 y_{max} + F_1 y(k) + F_2 \tilde{L}_w \Delta w_p \\ \tilde{L}_u \Delta u_f & \leq & F_1 \Delta y_{max} - \tilde{L}_w \Delta w_p \\ -\tilde{L}_u \Delta u_f & \leq & -F_1 \Delta y_{max} + \tilde{L}_w \Delta w_p \end{array} \tag{2.157}$$

where $u_{f_{max}} = F_3 u_{max}$, $u_{f_{min}} = F_3 u_{min}$ and $y_{f_{max}} = F_1 y_{max}$, $y_{f_{min}} = F_1 y_{min}$, and $\Delta u_{f_{max}}$, $\Delta u_{f_{min}}$, $\Delta y_{f_{max}}$ and $\Delta y_{f_{min}}$ are defined in a similar way. F_3 and F_4 are defined as,

$$F_3 = \begin{bmatrix} I_m \\ I_m \\ \vdots \\ I_m \end{bmatrix} \quad , \quad F_4 = \begin{bmatrix} I_m & 0 & \dots & 0 \\ I_m & I_m & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I_m & I_m & \dots & I_m \end{bmatrix} \tag{2.158}$$

These constraints in (2.157) are written in a single inequality as follows [21, 68],

$$\begin{bmatrix} F_4 \\ -F_4 \\ I_m \\ -I_m \\ F_2\tilde{L}_u \\ -F_2\tilde{L}_u \\ \tilde{L}_u \\ -\tilde{L}_u \end{bmatrix} \Delta u_f \leq \begin{bmatrix} F_3 u_{max} - F_3 u(k) \\ -F_3 u_{min} + F_3 u(k) \\ F_3 \Delta u_{max} \\ -F_3 \Delta u_{min} \\ F_1 y_{max} - F_1 y(k) - F_2 \tilde{L}_w \Delta w_p \\ -F_1 y_{max} + F_1 y(k) + F_2 \tilde{L}_w \Delta w_p \\ F_1 \Delta y_{max} - \tilde{L}_w \Delta w_p \\ -F_1 \Delta y_{max} + \tilde{L}_w \Delta w_p \end{bmatrix} \tag{2.159}$$

$$A_{QP} \Delta u_f \leq B_{QP} \tag{2.160}$$

Therefore, the constrained SPC problem is solved by considering the following QP formulation [21, 68],

$$\begin{aligned} \min_{\Delta u_f} \quad & J = \Delta u_f^T H \Delta u_f + P^T \Delta u_f \\ \text{subject to} \quad & A_{QP} \Delta u_f \leq B_{QP} \end{aligned} \tag{2.161}$$

The optimization can be done by using any of the standard commercially available QP optimization code at each sample time. Then the calculated control signal $u(k+1)$ is applied to the system. Constrained SPC has more computational load than unconstrained SPC. However, capability of constraint handling is one of the most attractive features of the SPC in industrial applications, since all of the real processes has some practical limitations on their parameters.

2.3 Subspace-based Approach for a Priori Knowledge Extraction

SIM is a data-driven black-box approach, but it has the capability to determine or estimate some of the system characteristics, such as system order, impulse response, step response, static gain and time-delay. These characteristics are derived directly from I/O data and can be used as a priori knowledge when they are required.

Impulse/Step Response Estimation

Recall the Subspace Predictor equation in (2.129),

$$\hat{Y}_f = L_w W_p + L_u U_f \qquad (2.162)$$

it has been shown that assuming that the number of row blocks, M in data block-Hankel matrices is large enough ($M \to \infty$) following result is obtained [18, 21],

$$L_u = H_M^d \qquad (2.163)$$

where L_u is the subspace predictor coefficient matrix which is calculated in (2.133) from the I/O data block-Hankel matrices by solving the Least Square problem in (2.130), and H_M^d is a lower triangular block Toeplitz matrix in (2.58). Considering the matrix H_M^d from (2.58), it can be seen that first column of H_M^d contains the *true* impulse response coefficients (IRCs) (Markov parameters) of the system for M horizon.

$$\bar{h}_M = H_M^d(:,1) = \begin{bmatrix} D \\ CB \\ CAB \\ \vdots \\ CA^{M-2}B \end{bmatrix} \qquad (2.164)$$

Moreover, *true* step response coefficients (SRCs) for M horizon are obtained as,

$$\bar{g}_M = F_2 H_M^d(:,1) = \begin{bmatrix} D \\ D + CB \\ D + CB + CAB \\ \vdots \\ D + CB + CAB + \cdots + CA^{M-2}B \end{bmatrix} \qquad (2.165)$$

where F_2 is defined as (2.144). Therefore, first M samples of IRCs and SRCs of the system can be estimated from the first column of matrix L_u as below,

$$\begin{aligned} \hat{h}_M &= L_u(:,1) && (2.166) \\ \hat{g}_M &= F_2 L_u(:,1) && (2.167) \end{aligned}$$

Time-delay and Static Gain Estimation

By estimating the IRCs and SRCs from (2.166) and (2.167), if there is a time-delay of T_d in I/O data (or N_d in terms of samples) the first N_d elements of the $\hat{h}_M$ and $\hat{g}_M$ must be approximately zero. Therefore, the time-delay can be estimated from (2.166) and (2.167) by applying an appropriate threshold. Moreover, static gain of the system and its sign can be determined from the estimated SRCs in (2.167).

Order Estimation

Consider the Subspace Predictor equation in (2.129). In this equation L_u and L_w are subspace predictor matrices, which are calculated in (2.133) from the I/O data block-Hankel matrices by solving the Least Square problem in (2.130) via QR decomposition in (2.131). From the equation (2.133), L_w can be obtained as,

$$L_w = (R_{31} - R_{32}R_{22}^{-1}R_{21})R_{11}^{-1} \tag{2.168}$$

In SIM framework, order of system can be estimated from the Singular Value Decomposition (SVD) of subspace predictor coefficient matrix L_w [62],

$$L_w = [U_1 \quad U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}^{\dagger} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \tag{2.169}$$

In practical applications L_w is not a rank deficient matrix, but it can be approximated as,

$$L_w \approx U_1 \Sigma_1^T V_1^T \tag{2.170}$$

which is a rank deficient matrix. Consequently, order of the system is estimated by applying an appropriate threshold and determining the number of dominant singular values, Σ_1 in (2.169) [62].

Moreover, it was shown that the following SVD can give better estimation of the system order than (2.168) [20] [106],

$$R_{31} - R_{32}R_{22}^{-1}R_{21} = [U_1 \quad U_2] \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix}^{\dagger} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \tag{2.171}$$

the order n will be the number of dominant singular values S_1 in (2.171).

2.4 Fuzzy Gain-Scheduling Procedure (FGS)

Most industrial processes are nonlinear and time-varying systems, their dynamic behavior changes with the operating conditions of the process. A commonly used scheme to deal with this issue is applying a *Gain Scheduling* (GS) technique [99]. In GS technique the controller parameters are updated by monitoring different operating conditions of the process. GS technique was first introduced in about 1950s with application on flight control systems [99], and it is very useful technique to reduce the effects of parameter variation in different processes, such as pH control, fuel-air control in car engine and ship steering [99]. Since the parameter updating procedure is done based on open-loop or pre-programmed way, it is controversial to consider the GS technique as an *adaptive* system. However, by utilizing and evaluating some auxiliary scheduling variables, which are correlated with the changes in system dynamics the GS technique can be considered as an adaptive controller [99]. In this case there is no need to estimate and identify the system parameters, because the controller parameters can be updated quickly by using the auxiliary scheduling variable values, as soon as a changes are observed in system dynamics. Figure 2.6 shows a typical block diagram of a system with GS technique.

Generally, a conventional GS procedure has following designing steps [99]:

1- Determine the suitable auxiliary scheduling variables based on the physical system knowledge.

2- Linearized the non-linear system at about different operating points. These linearized models are utilized to compute the local controller parameters for each of the operating conditions.

3- Apply the GS technique based on measurements of the auxiliary scheduling variables, which is done by monitoring different operating conditions.

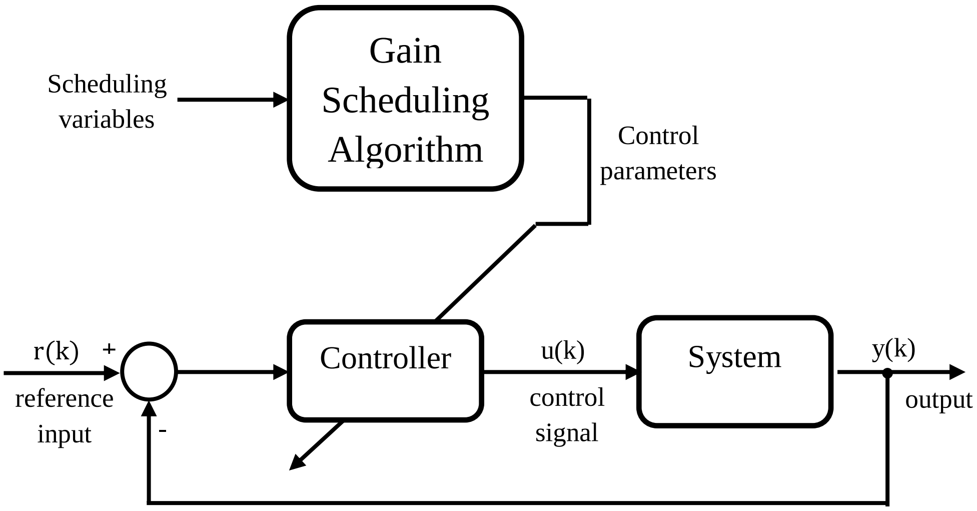

Figure 2.6: Typical structure of GS technique

The main issue in design of GS system is finding suitable auxiliary scheduling variables, which is done in model-based approach. Another drawback of conventional GS technique is that it is based on open-loop compensation method, and if the linearization is not accurate enough, the error can result to closed-loop instability. Moreover, the controller parameters must be determined for many operating points, which is very time consuming. Stability and local performance assessment of each linearized controller must be done for each operating point, and non-local performance evaluation is checked by simulations.

On the other hand, there is a fuzzy logic based approach to GS technique which can overcome the disadvantages of conventional GS technique. The approach is called *Fuzzy Gain-Scheduling* (FGS) technique. There are many successful implementation of FGS in the literature to control the nonlinear and time-varying systems [107–110], and FGS-PID, which is one of the most popular FGS techniques in industry [111–114].

In general, FGS procedure consist of three main parts *Fuzzifier*, *FGS Rules* and *Defuzzifier* as shown in Figure 2.7.

In FGS technique, first, the crisp input variables are converted to fuzzy numbers in *Fuzzifier*, then they are utilized to determine updated controller gains by considering *FGS Rules*. Finally, the resultant fuzzy numbers representing the controller gains are converted to crisp values by *Defuzzifier*.

A well-known issue in FGS technique is predefined the prescribed ranges of controller parameters. Since the controller parameters in FGS algorithm is considered in normalized format in range

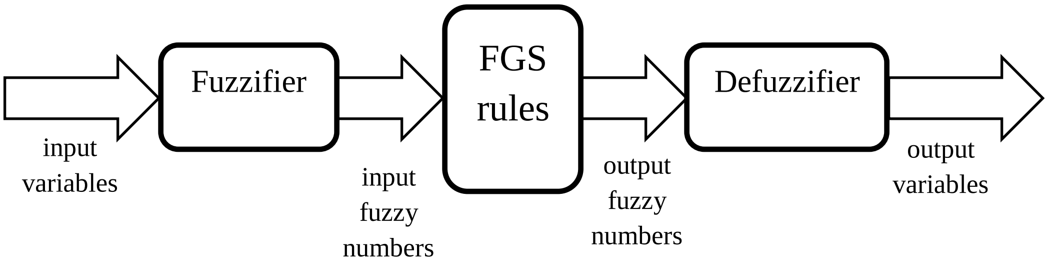

Figure 2.7: Typical structure of FGS

of zero to one, the output of defuzzifier block should be in the normalized form. Therefore, without having these controller gain ranges (CGRs) recovering the actual values of controller parameters is impossible. Currently, these CGRs are calculated in an ad hoc manner by the designer, which is a complex procedure especially for time-varying and nonlinear systems. The issue was addressed in the literature, and several methods were presented to determine the CGRs in FGS-PID controller [112, 115–118]. For example, in [117] CGRs are determined in FGS-PID by rule of thumb based on extensive simulation studies and by using the gain and period of oscillation at the stability limit under proportional control. The CGRs replaced by proper adjustment rates in [112] to generate the degrees of gain difference. However, the rates still need to be selected by the user according to the several simulation results of FGS-PID. In [116] CGRs were considered as scaling factors for a PID-type fuzzy controller, and a self-tuning method was presented to tune them via tracking error based functions. However, the simulation results show a considerable overshoot at the transient response of the controlled system. An optimal FGS-PID control is also presented in [115] to enhance the performance of conventional PID controller. The method uses Bee Colony Optimization technique to automatically tuning the scaling factors, membership functions and control rules of the FGS-PID controller automatically. However, on-line tuning of all membership functions and control rules is a complex task and demands high computational load. In [118] a Particle Swarm Optimization (PSO) based strategy was provided for tuning scaling factors in a PID-type fuzzy logic controller, by considering a cost function to minimize the maximum overshoot and the integral of absolute error. The approach was compared with standard Genetic Algorithm (GA) approach which shows computational efficiency and converges superiority of the PSO-based algorithm.

2.5 Particle Swarm Optimization Technique (PSO)

PSO is an intelligent based optimization technique, and belongs to meta-heuristic class of optimization algorithms [119–121], which is inspired by social behavior of animals like ants, fish and birds. It has superiority to other meta-heuristic optimization methods such as GA in terms of complexity and computational load [118, 122]. There are many successful engineering applications of PSO in the literature [123, 124]. PSO is a population based technique which contains a swarm of candidate solutions called *particles*. Each particle has a *position* in a multidimensional *search space* of the optimization problem. The search space contains all possible solutions for the optimization problem. PSO searches for the optimum solution among these possible solutions.

Mathematical model for motion of particles and updating their position in PSO is described as [121],

$$\begin{aligned} v_i(t+1) &= w(t)v_i(t) + r_1c_1(pb_i(t) - p_i(t)) + r_2c_2(pg(t) - p_i(t)) \qquad (2.172) \\ p_i(t+1) &= p_i(t) + v_i(t+1) \qquad (2.173) \end{aligned}$$

where, i is the index of the particle, t is the discrete time step and shows the iteration of the algorithm, $p_i(t)$ and $v_i(t)$ are position and velocity of ith particle in the search space at time step t, respectively. $w(t)$ is the inertia coefficient. r_1 and r_2 are uniformly distributed random numbers between zero and one. c_1 and c_2 are positive acceleration coefficients to scale the cognitive component $(pb_i(t) - p_i(t))$ and the social component $(pg(t) - p_i(t))$, respectively. $pb_i(t)$ is the personal best position for ith particle and $pg(t)$ is the global best position among the members of swarm at iteration t. The personal best position and the global best position, at the next iteration $t + 1$ are updated as follows [121],

$$pb_i(t+1) = \begin{cases} pb_i(t) & \text{if } J(p_i(t+1)) \geq J(pb_i(t)) \\ p_i(t+1) & \text{if } J(p_i(t+1)) < J(pb_i(t)) \end{cases} \qquad (2.174)$$

$$pg(t+1) = \begin{cases} pg(t) & \text{if } J(pb_i(t+1)) \geq J(pg(t)) \\ pb_i(t+1) & \text{if } J(pb_i(t+1)) < J(pg(t)) \end{cases} \qquad (2.175)$$

where $J(p_i(t+1))$ is the PSO fitness function evaluation for particle i at the iteration $t + 1$.

Suitable selection of inertia coefficient, $w(t)$, is essential to ensure convergence behavior of optimization problem. The inertia coefficient has effect on narrowing the search area gradually by changing the exploratory mode to an exploitative mode. In practical applications, w normally decreases from $w_{max} = 0.9$ to $w_{min} = 0.4$ from the beginning to the end of the optimization problem [121]. In each iteration the inertia coefficient is calculated as follows,

$$w(t) = (w_{min} - w_{max})\frac{IT(t)}{IT_{max}} + w_{max} \tag{2.176}$$

where, $IT(t)$ and IT_{max} are current iteration number and maximum number of iterations, respectively. Since, PSO is an iterative optimization procedure a convergence criteria is required to terminate the iteration. Some of these terminating conditions are such as [125]: Exceeding the maximum number of iteration, finding an acceptable solution, observing no improvement over a number of iteration, achieving to almost zero normalized swarm radius or achieving approximately zero slope in objective function. General principle of PSO algorithm is shown in Figure 2.8.

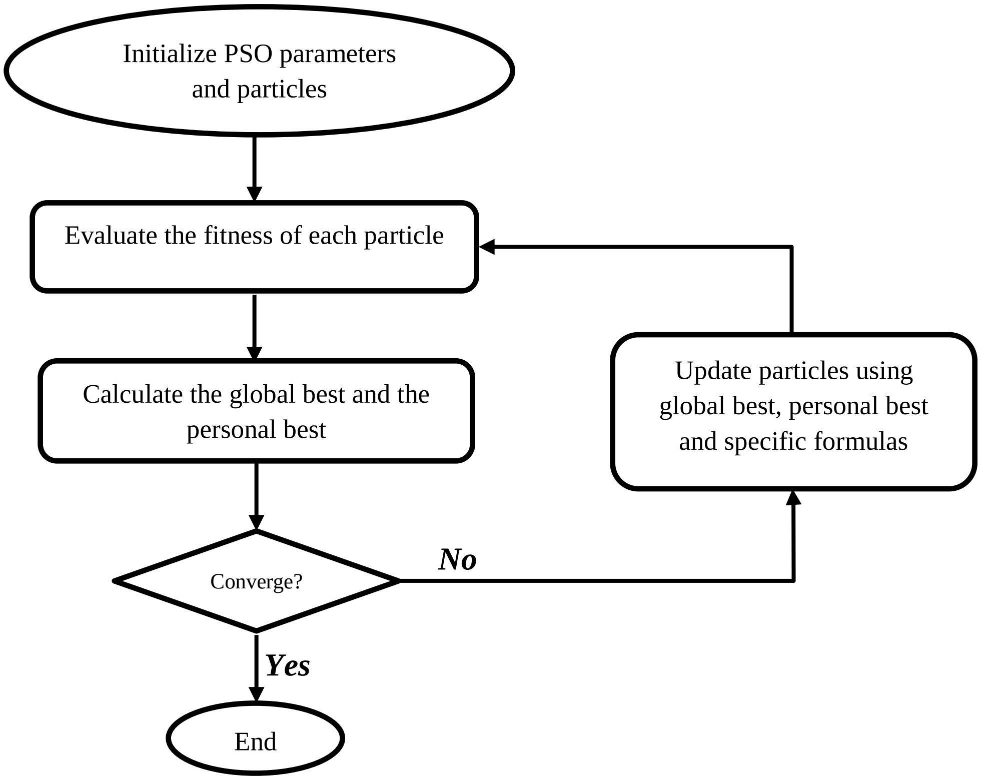
Initialize PSO parameters and particles
Evaluate the fitness of each particle
Calculate the global best and the personal best
Update particles using global best, personal best and specific formulas
Converge?
No
Yes
End

Figure 2.8: Basic principle of PSO

A more generalized PSO algorithm was suggested in [120], which can improve the convergence quality by decreasing the amplitude of the particle movements as they approach to the solution.

This method is also called *Constriction PSO*, which is formulated as follows,

$$\begin{aligned} v_i(t+1) &= \chi[v_i(t) + r_1\phi_1(pb_i(t) - p_i(t)) + r_2\phi_2(pg(t) - p_i(t))] && (2.177) \\ p_i(t+1) &= p_i(t) + v_i(t+1) && (2.178) \end{aligned}$$

where $\phi_1 = \phi_2 = 2.05$ and χ is calculated as,

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \tag{2.179}$$

where $\phi = \phi_1 + \phi_2$ and $\kappa = 1$. Parameter χ guaranties a decreasing velocity for each particle by increasing the number of iterations [120, 126].

2.6 Summary

In this chapter, an introduction to system identification methods and predictive control were provided. Focusing on open-loop system identification, the Prediction Error Method (PEM) and the Subspace Identification Methods (SIM) were reviewed. A review of model-based approach and data-driven approach of predictive control were presented, and formulation of Model Predictive Control (MPC) and Subspace Predictive Control (SPC) were provided for both unconstrained and constrained cases. A review of subspace-based methods to estimate the impulse response and step response, static gain, time-delay and system order were also included. Moreover, a review of Fuzzy Gain Scheduling (FGS) technique and Particle Swarm Optimization (PSO) procedure were presented.

Chapter 3

Stability and Performance of SPC

3.1 Problem Statement and Chapter Summary

In SPC implementation, the SPC gain matrices and control signal are functions of SPC tuning parameters. Therefore, appropriate choice of SPC tuning parameters, horizons and weighting matrices, have a significant impact on stability and performance of SPC closed-loop system. Moreover, the SPC gains are calculated directly by using I/O data, and computational complexity of the optimization algorithm depends on the dimension of the system, which is also affected by the length of SPC horizons. Therefore, it is required to select the SPC horizons in an optimal way to guarantee the closed-loop stability and performance and to minimize the computational load. There exist tuning strategies and stability analysis for MPC, which are based on model-based techniques [48]. These methods need some prior information from the system model and parameters, such as open-loop settling time, rise time, delay time or system matrices and only some of them can guarantee the closed-loop stability. However, SPC is a model-free approach, and these model-based techniques are not applicable for SPC. In the best of our knowledge, there is no model-free approach for tuning the SPC horizons. In SPC approach the controller is designed based on subspace predictor matrices L_u and L_w, which are driven directly from I/O data. Therefore, it is necessary to provide an *optimal model-free tuning technique* for SPC based on subspace predictor matrices that can also guarantee the closed-loop stability. Here, we concentrate on open-loop stable systems, but the systems can be non-minimum phase.

In this chapter, SPC closed-loop stability conditions are provided for open-loop stable systems based on SPC gains and SPC cost function tuning parameters. Consequently, SPC stability graph is introduced and feasibility range of prediction horizon is determined to ensure the closed-loop stability. Moreover, a technique is provided to calculate the efficient prediction and control horizons to guarantee the closed-loop stability and performance and to minimize the computational load.

3.2 SPC Stability Analysis

Considering the SPC incremental control law from (2.147) and SPC gain matrices in (2.148) and (2.149), this section provides SPC closed-loop stability conditions based on the SPC gains and SPC cost function tuning parameters.

3.2.1 SPC Closed-loop Transfer Function

Assume that a SISO open-loop stable system is described by the following discrete time transfer function,

$$G_{ol}(q^{-1}) = \frac{y(k)}{u(k)} = \frac{N(q^{-1})}{D(q^{-1})} \tag{3.1}$$

where $N(q^{-1})$ and $D(q^{-1})$ are polynomials with unknown coefficients. Note that the system can be non-minimum phase with time-delay.

On the other hand, by considering Δw_p in (2.143) the SPC control law in (2.150) can be rewritten as below,

$$\Delta u(k+1) \quad = \quad -\tilde{k}_e(y(k) - r(k+1)) - \tilde{k}_{\Delta y_p}(q^{-1})\Delta y(k) - \tilde{k}_{\Delta u_p}(q^{-1})\Delta u(k) \tag{3.2}$$

where $\tilde{k}_{\Delta y_p}(q^{-1})$ and $\tilde{k}_{\Delta u_p}(q^{-1})$ are defined as,

$$\begin{aligned} \tilde{k}_{\Delta y_p}(q^{-1}) &= \tilde{k}_{\Delta y_{p_M}} q^{-M+1} + \cdots + \tilde{k}_{\Delta y_{p_2}} q^{-1} + \tilde{k}_{\Delta y_{p_1}} && (3.3) \\ \tilde{k}_{\Delta u_p}(q^{-1}) &= \tilde{k}_{\Delta u_{p_M}} q^{-M+1} + \cdots + \tilde{k}_{\Delta u_{p_2}} q^{-1} + \tilde{k}_{\Delta u_{p_1}} && (3.4) \end{aligned}$$

See Appendix A for calculations.

By multiplying both sides of (3.1) by $\Delta = 1 - q^{-1}$ and using (3.2) the *SPC closed-loop transfer function* is obtained as follows,

$$G_{cl}(q^{-1}) \quad = \quad \frac{y(k)}{r(k+1)} = \frac{q^{-1}N(q^{-1})}{D_{cl}(q^{-1})} \tag{3.5}$$

where $D_{cl}(q^{-1})$ is the *closed-loop characteristics polynomial* as below,

$$D_{cl}(q^{-1}) = \Delta\tilde{k}_e^{-1}D(q^{-1}) + q^{-1}[\Delta\tilde{k}_e^{-1}N(q^{-1})\tilde{k}_{\Delta y_p}(q^{-1}) + \Delta\tilde{k}_e^{-1}D(q^{-1})\tilde{k}_{\Delta u_p}(q^{-1}) + N(q^{-1})] \quad (3.6)$$

See Appendix B for calculations.

Remark 3.2.1. *In MIMO systems with m inputs and l outputs, the transfer function matrix can be considered as follows,*

$$y(k) = G_{ol}(q^{-1})u(k) \quad (3.7)$$

where $y(k) \in \mathbb{R}^l$, $u(k) \in \mathbb{R}^m$ and $G_{ol}(q^{-1})$ is defined as below,

$$G_{ol}(q^{-1}) = \begin{bmatrix} G_{11} & G_{12} & \dots & G_{1m} \\ G_{21} & G_{22} & \dots & G_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ G_{l1} & G_{l2} & \dots & G_{lm} \end{bmatrix} \quad (3.8)$$

where G_{ij} is transfer function from jth input to ith output.

3.2.2 SPC Closed-loop Stability

The SPC closed-loop characteristics equation in (3.6) can be rewritten as follows,

$$\Delta\tilde{k}_e^{-1}D(q^{-1}) + N^*(q^{-1}) = 0 \quad (3.9)$$

where $N^*(q^{-1})$ is defined as,

$$N^*(q^{-1}) \quad = \quad q^{-1}[\Delta\tilde{k}_e^{-1}N(q^{-1})\tilde{k}_{\Delta y_p}(q^{-1}) + \Delta\tilde{k}_e^{-1}D(q^{-1})\tilde{k}_{\Delta u_p}(q^{-1}) + N(q^{-1})] \quad (3.10)$$

and the closed-loop poles are also satisfying the following equation,

$$1 + \frac{\tilde{k}_e}{\Delta}\frac{N^*(q^{-1})}{D(q^{-1})} = 0 \quad (3.11)$$

Structure of (3.11) indicates that the SPC closed-loop characteristics equation can be defined as a negative feedback control system, which includes a *compound system* with transfer function of,

$$\frac{N^*(q^{-1})}{D(q^{-1})} \tag{3.12}$$

and an *integral type regulator* with transfer function of,

$$\frac{\tilde{k}_e}{\Delta} \tag{3.13}$$

By considering that $\Delta(1) = 0$, DC-gain of the compound system in (3.12), can be derived as,

$$K_{co} \quad = \quad \frac{N^*(1)}{D(1)} = \frac{N(1)}{D(1)} = K_{ol} \tag{3.14}$$

where K_{co} is DC-gain of the compound system in (3.12) and K_{ol} is DC-gain of the open-loop system in (3.1).

Theorem 3.2.1. *The necessary and sufficient condition for small-gain stability of SPC closed-loop system in (3.5) is,*

$$\tilde{k}_e K_{ol} > 0 \tag{3.15}$$

where $\tilde{k}_e$ and K_{ol} are defined in (2.150) and (3.14), respectively.

Proof. Applying similar small-gain root-locus analysis as in [127] and [128] to (3.11) proves stability of the SPC closed-loop system [129]. See Appendix C for proof.

□

Remark 3.2.2. *Note that in (3.15), only sign of the open-loop DC-gain, sgn$[K_{ol}]$, is required, which can be determined by the method in Section 2.3.*

Remark 3.2.3. *Since, $\tilde{k}_e$ is defined as the first m rows of $\tilde{K}_e$, recall (2.148) shows that $\tilde{k}_e$ is proportional to the weighting matrix Q and inversely proportional to the weighting matrix R. Therefore, the concept of small-gain $\tilde{k}_e$ can be interpreted as sufficiently large value of R and sufficiently small value of Q.*

Remark 3.2.4. *Given sufficiently small $Q \geq 0$ and sufficiently large $R > 0$, the SPC closed-loop system defined in (3.5) is stable, if N_p and N_c ($N_p > N_c$) are selected such that the SPC closed-loop* *stability condition in (3.15) is satisfied.*

Definition 3.2.1. *SPC gains that satisfy (2.148) and (2.149) are called **achievable SPC gains**, and the achievable SPC gains which ensure SPC closed-loop stability in (3.15) are called **feasible SPC gains**. Similar definitions are used for MPC in [49].*

3.3 SPC Tuning Parameters Selection

The SPC cost function in (2.117) has four tuning parameters: Weighting matrices Q, R, Prediction horizon N_p and Control horizon N_c. Appropriate selection of them has significant effect on stability and performance of the SPC controller. Weighting matrices have similar roles and characteristics as in Linear Quadratic (LQ) controllers [48]. However, prediction horizon N_p and control horizon N_c are the specific parameters for SPC, as LQ controllers use infinite horizon in the optimization algorithm. This section analyzes the effect of selecting SPC tunning parameters on stability and performance of the closed-loop system.

3.3.1 Weighting Matrices (Q and R)

In SPC cost function weighting matrices Q and R are considered as positive semidefinite and positive definite matrices to penalize tracking error and control signal, respectively. Weights on the tracking error, Q, are used to direct more control effort towards more dominant output signal. Setting larger weight for a particular output leads to a faster, but aggressive response for that particular output. Weighting matrix R is utilized to penalize the control signal. Setting a large penalty for a specific control signal yields a more robust, but slow controller. On the other hand, smaller weights result in more aggressive, but less robust controllers [48].

3.3.2 Prediction Horizon ($\mathbf{N_p}$)

Prediction horizon is considered as the number of future samples needed to be computed to minimize tracking error. N_p is chosen long enough to ensure that the controlled system output reaches steady state. For practical applications, N_p needs to be larger than the control horizon and is chosen at least equal to the open-loop settling time [48, 54]. Good stability properties has been observed by using a large prediction horizon, which leads to a control law that approaches LQ control [61, 130]. However, the use of large N_p does not always guarantee the closed-loop stability and may cause numerical instability, and also drastically increases the computational load especially for systems with fast dynamics. On the other hand, selecting very short prediction horizon can jeopardize the robustness and stability of the SPC closed-loop system.

In SPC approach selecting the prediction horizon N_p truncates the rows of subspace predictor matrices $L_w \in \mathbb{R}^{Ml \times M(l+m)}$ and $L_u \in \mathbb{R}^{Ml \times Mm}$, as shown in (2.139) and (2.140). Therefore, N_p cannot be considered larger than M. By considering $N_c < N_p$, the *achievable prediction horizon range* for the SPC approach is defined as,

$$1 \leq N_c < N_p \leq M \tag{3.16}$$

However, not all of the achievable prediction horizons can guarantee the SPC closed-loop stability. Therefore, the feasible range of prediction horizon has to be defined based on SPC closed-loop stability condition in (3.15).

Definition 3.3.1. *The achievable range of N_p that guarantees the SPC closed-loop stability condition in (3.15) is called the* ***Feasible Range (FR) of prediction horizon****.*

Definition 3.3.2. *The smallest prediction horizon that ensures the SPC closed-loop stability condition in (3.15) is called* ***Shortest-Feasible-Prediction-horizon (SFP)*** *and is denoted by N_{SFP}.*

Determine $\mathbf{N_{SFP}}$ and FR of Prediction Horizon

Table 3.1 provides a technique to obtain the feasible SPC gains based on the SPC stability condition in (3.15) and Remark 3.2.4 for given weighting matrices, $Q \geq 0$ and $R > 0$, then to determine the N_{SFP} and FR of N_p [129].

Table 3.1: Technique of determining the feasible SPC gains, N_{SFP} and FR of N_p

Step 1: Determine sgn$[K_{ol}]$ using open-loop I/O data

Step 2: Given Q and R calculate achievable SPC gains $\tilde{k}_e$, for $1 \leq N_c < N_p \leq M$ and $N_c = N_p - 1$

Step 3: Obtain the feasible SPC gains $\tilde{k}_e$ by using SPC closed-loop stability condition in (3.15), and determine N_{SFP} and FR of N_p

Remark 3.3.1. *According to the SPC stability condition in (3.15) and the technique in Table 3.1, the **Efficient-Prediction-horizon**, N_{EP} must be selected as,*

$$N_{SFP} \leq N_{EP} \leq M \tag{3.17}$$

Moreover, choice of N_{EP} depends on the desired closed-loop performance for each system. Here, we propose optimal selection of N_{EP} based on the SPC cost function optimization. The strategy is provided in Table 3.2.

3.3.3 Control Horizon ($\mathbf{N_c}$)

Control horizon determines behavior of the control signal. Selecting a very small control horizon can be computationally more efficient, but deteriorates performance of the closed-loop system [48]. In addition, increasing the control horizon from a certain value has no significant effect on the closed-loop performance, but increases the computational load of the optimization algorithm and closed-loop system [48, 56].

In SPC approach selecting the control horizon, N_c, truncates the columns of subspace predictor matrix $L_u \in \mathbb{R}^{Ml \times Mm}$, which is shown in (2.140). The *achievable control horizon range* for SPC approach is defined as,

$$1 \leq N_c < N_p \tag{3.18}$$

It can be shown that for sufficiently small $Q \geq 0$ and sufficiently large $R > 0$, control horizon does not affect the FR of N_p and SPC closed-loop stability. However, N_c has an impact on the performance of the closed-loop system.

Selection of Efficient Control Horizon and Prediction Horizon ($\mathbf{N_{EC}}$ and $\mathbf{N_{EP}}$)

In the subspace identification literature, by assuming that the number of row blocks, M, in the data block Hankel matrices is large enough ($M \to \infty$), the subspace predictor coefficient matrix, L_u in (2.135) is considered as the estimate of matrix H_M^d [18, 21]. Recall the matrix H_M^d in (2.58), it can be seen that H_M^d is a lower triangular block Toeplitz matrix which contains the impulse response coefficients (Markov parameters) of deterministic inputs. In (2.140) it has been shown that selecting the control horizon truncates the columns of subspace predictor coefficient matrix L_u. Truncating the columns of matrix L_u is equivalent to rank reduction of matrix H_M^d. To represent an

nth order system properly by its Markov parameters, the number of parameters has to be selected large enough, and it is necessary to have at least n Markov parameters to represent the system. Therefore, for an observable and reachable nth order system it needs to have,

$$\min\{\ rank(H_M^d)\} = n+1 \tag{3.19}$$

Therefore, in truncating the columns of matrix L_u, control horizon has to be chosen greater than or equal to $n+1$, ($N_c \geq n+1$). Choice of $N_c < n+1$ can deteriorate the closed-loop performance. Moreover, it can be shown that increasing the N_c greater than N_{SFP} does not have significant effect on the feasible SPC gain $\tilde{k}_e$ and the performance. However, it reduces efficiency of the control algorithm by increasing the computational load.

Proposition 3.3.1. *In SPC approach for closed-loop system in (3.5), to avoid performance deterioration, and for computational efficiency, the* ***Efficient-Control-horizon****,* N_{EC}*, is suggested to be selected as follows by efficiently minimizing dimension of the subspace predictor matrix* L_u *[129],*

$$\begin{cases} N_{EC} = n+1 & if \quad N_{SFP} \leq n+1 \\ N_{EC} = N_{SFP} & if \quad N_{SFP} > n+1 \end{cases} \tag{3.20}$$

Remark 3.3.2. *Since, SPC is a model-free approach, the system order, n, can be estimated by applying Subspace-based order estimation method in Section 2.3. Full discussion is provided in Chapter 5.*

Table 3.2 provides an efficient methodology to select the SPC horizons based on the technique in Table 3.1, and definitions of N_{EC} in (3.20) and N_{EP} in (3.17).

Table 3.2: Technique of determining the efficient SPC horizons, N_{EC} and N_{EP}

Step 1: Given Q and R determine N_{SFP} and FR of N_p by using the technique in Table 3.1

Step 2: Determine N_{EC} based on the proposed criteria in (3.20).

Step 3: Given Q and R, determine N_{EP} by optimizing the SPC cost function in (2.118) as below,

$$\begin{aligned} \min_{N_p} \quad & J = (\hat{y}_f - r_f)^T Q (\hat{y}_f - r_f) + \Delta u_f^T R \Delta u_f \\ \text{subject to} \quad & \text{given } \ Q \geq 0, \ \ R > 0 \\ & N_c = N_{EC} \\ & N_{SFP} \leq N_p \leq M \\ & v_{min} \leq v \leq v_{max} \end{aligned}$$

where v is any of the constrained variables in (2.126).

3.4 Simulation Results

In this section, five examples are presented to demonstrate the proposed SPC stability condition by applying the technique in Table 3.1. Simulation results show the effectiveness of the proposed technique in Table 3.2 to select the efficient SPC horizons and stable SPC design. The selected systems have been extensively used in simulations to validate the controller performance [20, 21, 40, 45, 49, 53, 55, 131, 132].

3.4.1 Systems Description

System I: Non-minimum Phase Fourth-order System

For the first system an open-loop I/O dataset ($N = 2000$, $M = 60$ and $T_s = 0.5sec$) is collected from the following forth-order ($n = 4$) SISO non-minimum phase stable system [20],

$$A = \begin{bmatrix} 0.603 & 0.603 & 0 & 0 \\ -0.603 & 0.603 & 0 & 0 \\ 0 & 0 & -0.603 & -0.603 \\ 0 & 0 & 0.603 & -0.603 \end{bmatrix}, \quad B = \begin{bmatrix} 0.9238 \\ 2.7577 \\ 4.3171 \\ -2.6436 \end{bmatrix} \tag{3.21}$$

$$C = \begin{bmatrix} -0.5749 & 1.0751 & -0.5225 & 0.1830 \end{bmatrix}, \quad D = \begin{bmatrix} -0.7139 \end{bmatrix} \tag{3.22}$$

System II: Non-minimum Phase Second-order System

Here, open-loop I/O dataset with $N = 2000$, $M = 60$ and $T_s = 0.5sec$ is collected from the following second-order ($n = 2$) SISO high stable non-minimum phase system [45],

$$A = \begin{bmatrix} 1 & -0.25 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 1.2 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix} \tag{3.23}$$

System III: Non-minimum Phase Fourth-order System

In this case a fourth-order SISO non-minimum phase stable system is selected to collect open-loop I/O dataset for $N = 2000$, $M = 60$ and $T_s = 0.1sec$ [40],

$$A = \begin{bmatrix} 1.5310 & -0.6914 & 0.0995 & -0.0256 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.24}$$

$$C = \begin{bmatrix} -0.0876 & -0.1897 & 0.2947 & 0.0758 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix} \tag{3.25}$$

System IV: SISO time-delayed system

Consider following second-order plus time-delay system from [55],

$$G_{ol}(s) = \frac{e^{-50s}}{(150s+1)(25s+1)} \tag{3.26}$$

Sampling time of $T_s = 5sec$ is selected for System IV, and discretized transfer function of the system, by using zero order holder, is obtained as,

$$G_{ol}(z^{-1}) = z^{-11}\frac{0.003087 + 0.002856z^{-1}}{1 - 1.786z^{-1} + 0.7919^{-2}} \tag{3.27}$$

Open-loop I/O dataset is collected from the discretized system ($N = 2000$ and $M = 150$).

System V (Distillation System): MIMO time-delayed system

The Wood-Berry distillation column system is a Methanol-Water distillation column, which is a continuous time two-input two-output process with large time-delay. Transfer function of this MIMO model is [133],

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{16.7s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \tag{3.28}$$

In this system, y_1, y_2 are two outputs, which are the overhead product compositions and the bottom product composition, respectively. The outputs are controlled by manipulating the flux flow rate and steam flow rate as inputs, u_1 and u_2 [133]. Sampling time of $T_s = 1min$ is selected, and discretized transfer function matrix of the system, by using zero order holder, is obtained as follows,

$$G_p(z^{-1}) = \begin{bmatrix} z^{-2}\frac{0.744}{1-0.9419z^{-1}} & z^{-4}\frac{-0.8789}{1-0.9535z^{-1}} \\ z^{-8}\frac{0.5796}{1-0.9123z^{-1}} & z^{-4}\frac{-1.302}{1-0.9329z^{-1}} \end{bmatrix} \tag{3.29}$$

Open-loop I/O data for all inputs and outputs are collected for this discretized MIMO system ($N = 2000$, $M = 80$). Note that open-loop I/O data for all of the systems are obtained by using PRBS of magnitude 2 for the inputs $u(k)$.

3.4.2 Stability Analysis and Determination of $\mathbf{N_{SFP}}$ and FR of $\mathbf{N_p}$

FR of prediction horizon and N_{SFP} for each system are calculated by applying the technique in Table 3.1 as follows:

Step1: Sign of DC-gain, $sgn[K_{ol}]$, for each open-loop system is determine by using I/O data. The DC-gain signs are provided in Table 3.3 for SISO systems, and Table 3.4 shows the sign of static gains for each subsystem of the MIMO Distillation system. Note that, as shown in (3.15), only sign of the DC-gain is used not the exact value.

Step2: Given weighting matrices Q and R, the achievable SPC gains, $\tilde{k}_e$, are calculated for each

Table 3.3: Sign of DC-gain for open-loop SISO systems

System	$sgn[K_{ol}]$
I	-
II	+
III	+
IV	+

Table 3.4: Sign of DC-gain for each subsystem of Distillation System

Subsystem	sgn[DC-gain]
G_{p11}	+
G_{p21}	+
G_{p12}	-
G_{p22}	-

system for achievable prediction horizon range, $1 \leq N_c < N_p \leq M$, and $N_c = N_p - 1$.

Figures 3.1, 3.2, 3.3 and 3.4 show the achievable SPC gains, $\tilde{k}_e$, in terms of achievable prediction horizon range for each SISO system. We denoted these graphs as ***SPC stability graphs***. The SPC stability graphs for each subsystem of MIMO Distillation System are shown in Figure 3.5.

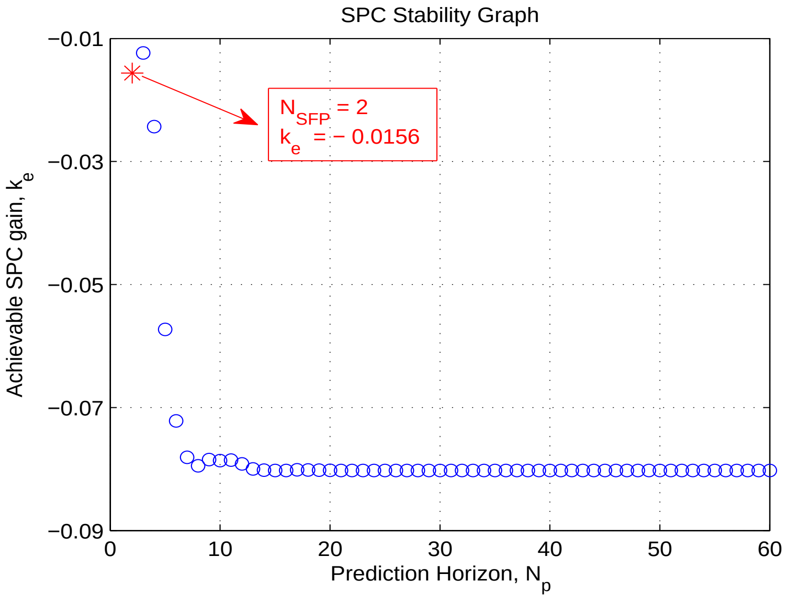

Figure 3.1: System I: SPC stability graph for $Q = I$, $R = 100I$ and $N_c = N_p - 1$

Remark 3.4.1. *Note that, for simplicity and without loss of generality, for all simulation the weighting matrices are selected as, $Q = I$ and $R = 100I$ for SISO systems, and $Q = 0.01I$ and*

$R = 400I$ for Distillation System, otherwise they can be chosen any positive semi-definite and positive definite matrices.

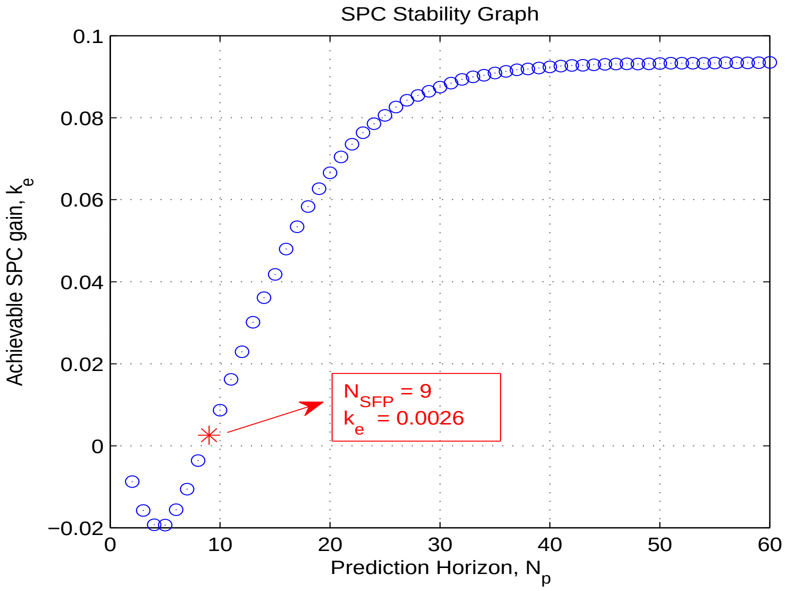

Figure 3.2: System II: SPC stability graph for $Q = I$, $R = 100I$ and $N_c = N_p - 1$

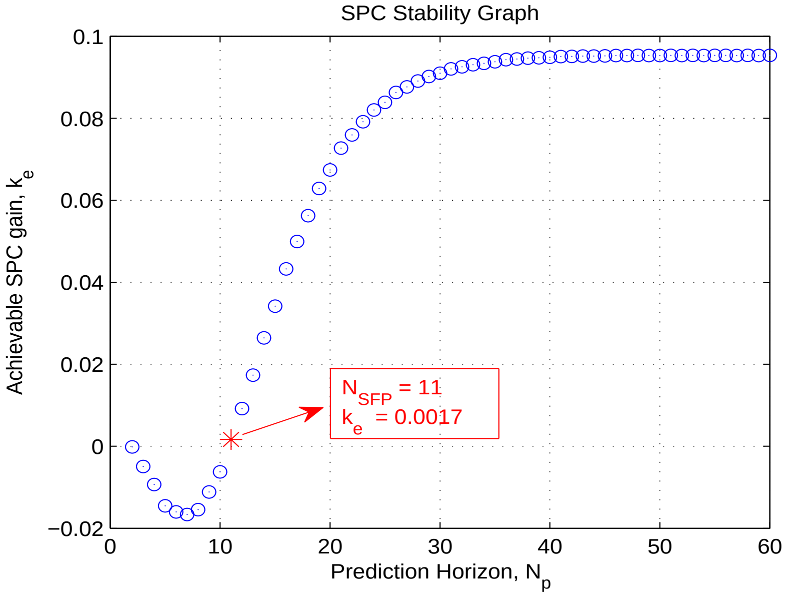

Figure 3.3: System III: SPC stability graph for $Q = I$, $R = 100I$ and $N_c = N_p - 1$

Step3: From the plotted SPC stability graphs in Figures 3.1, 3.2, 3.3 and 3.4, the N_{SFP} and FR of N_p are determined by comparing the $sgn[K_{ol}]$, and sign of achievable SPC gains, $sgn[\tilde{k}_e]$, for each

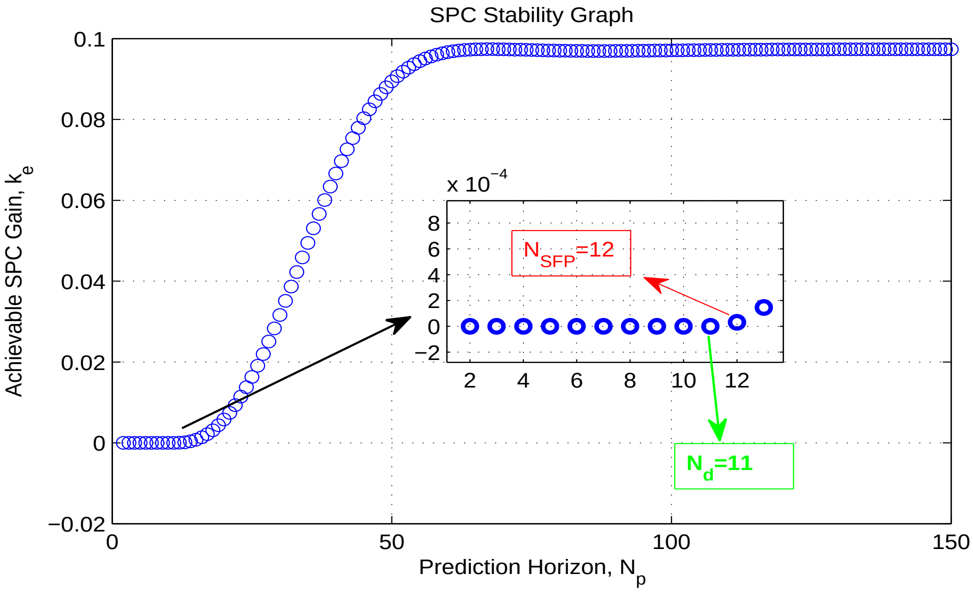

Figure 3.4: System IV: SPC stability graph for $Q = I$, $R = 100I$ and $N_c = N_p - 1$

SISO system, to satisfy the SPC closed-loop stability condition in (3.15). For MIMO Distillation System the sign of DC-gain of each subsystem must be considered. The N_{SFP} for each system and subsystems are shown in Figures 3.1, 3.2, 3.3, 3.4 and 3.5.

For System I, from Table 3.3 the DC-gain is negative, $K_{ol} < 0$, and according to the stability graph in Figure 3.1 all achievable SPC gains are negative, $\tilde{k}_e < 0$. Therefore, from the stability condition in (3.15), $\tilde{k}_e K_{ol} > 0$, all achievable SPC gains are feasible. As a result, FR of N_p for System I is equal to the achievable range of N_p, and $N_{SFP} = 2$.

However, for Systems II, III and IV, from Table 3.3, the DC-gains are positive, $K_{ol} > 0$, and the SPC stability graphs in Figures 3.2, 3.3 and 3.4 indicate that not all of the achievable SPC gains can satisfy the stability condition in (3.15). According to the stability graphs in Figures 3.2 and 3.3 the value of N_{SFP} for Systems II and III are $N_{SFP} = 9$ and $N_{SFP} = 11$, respectively. Therefore, the FR of N_p for Systems II and III can be obtained as $9 \leq N_p \leq M$ and $11 \leq N_p \leq M$, respectively. For System IV the time-delay can also be determined from SPC stability graph in noise-free cases. From SPC stability graph in Figure 3.4 we obtain $N_d = 11$, $N_{SFP} = 12$ and FR of N_p is $12 \leq N_p \leq M$. The results for each SISO system have been shown in Table 3.5.

For MIMO Distillation System, from the SPC stability graphs in Figure 3.5 and sign of the DC-gains in Table 3.4, N_{SFP} and FR of N_p are determined as Table 3.6 for each subsystem. Note that from Figure 3.5 it can be seen that in subsystem G_{p21} for $N_p > 38$ the SPC gain k_e becomes negative. Therefore, FR of N_p for subsystem G_{p21} will be $9 \leq N_p \leq 38$. Moreover, time-delay of

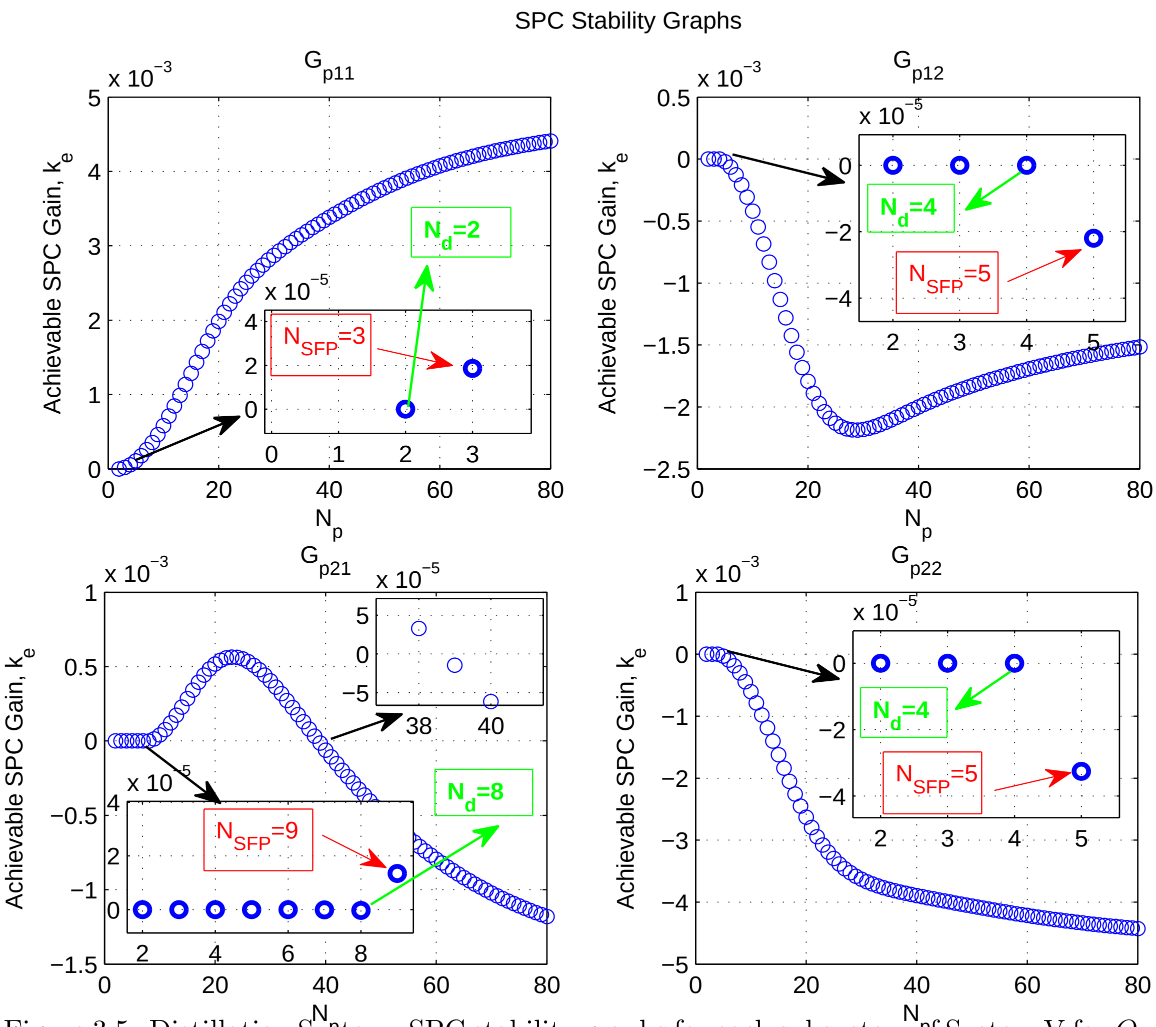

Figure 3.5: Distillation System: SPC stability graphs for each subsystem of System V for $Q = 0.01I$, $R = 400I$ and $N_c = N_p - 1$

Table 3.5: N_{SFP} and FR of prediction Horizon for each system

System	N_{SFP}	FR of N_p
I	2	$2 \leq N_p \leq M$
II	9	$9 \leq N_p \leq M$
III	11	$11 \leq N_p \leq M$
IV	12	$12 \leq N_p \leq M$

each subsystem is also detectable from the SPC stability graphs.

Figure 3.6 shows output of the System III for different feasible and unfeasible N_p values. From Table 3.5 the feasible range of prediction horizon for System III is $11 \leq N_p \leq M$. Therefore, Figure 3.6 indicates that by selecting the N_p from the feasible range of prediction horizon an stable

Table 3.6: N_{SFP} and FR of N_p for Distillation System ($M = 80$)

Subsystem	N_{SFP}	FR of N_p
G_{p11}	3	$3 \leq N_p \leq M$
G_{p21}	9	$9 \leq N_p \leq 38$
G_{p12}	5	$5 \leq N_p \leq M$
G_{p22}	5	$5 \leq N_p \leq M$

closed-loop system is achievable. However, for unfeasible prediction horizon values, $N_p = 5$ and $N_p = 10$, the SPC closed-loop system is unstable.

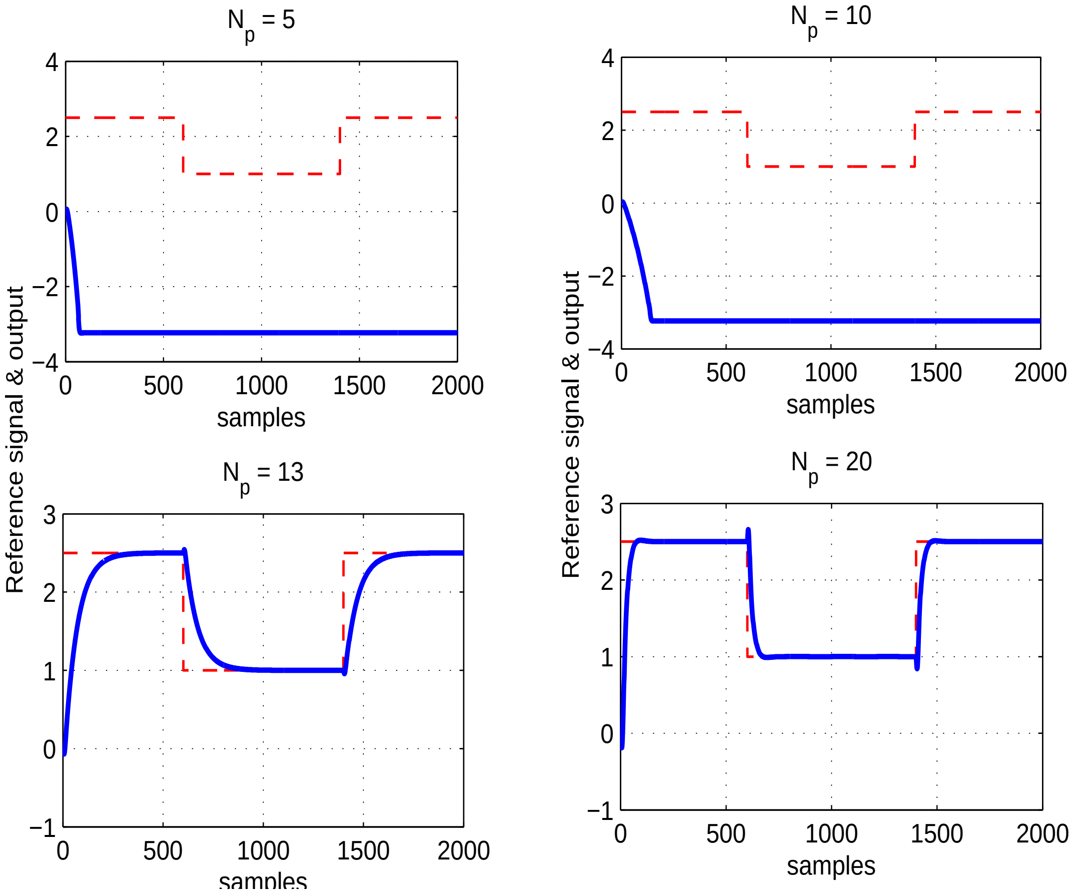

Figure 3.6: System III: Reference signal and output for $Q = I$, $R = 100I$, $N_c = N_p - 1$ and different unfeasible and feasible N_p values

3.4.3 Choice of Efficient SPC Horizons $\mathbf{N_{EC}}$ and $\mathbf{N_{EP}}$

By determining the FR of N_p the optimum SPC horizons need to be determined from the feasible values. In this part the efficient SPC horizons, N_{EC} and N_{EP}, are determined for each system based on the technique in Table 3.2 as follows:

Step 1: The N_{SFP} and FR of N_p are determined for each system by using the technique in Table 3.1. The results are shown in Tables 3.5 and 3.6.

Step 2: The efficient-control-horizon, N_{EC}, can be determined for each system from the proposed criteria in (3.20). For this purpose order of each system needs to be compared with its N_{SFP}. The results are shown in Table 3.7 for each system.

Remark 3.4.2. *Note that the system order, n, can be estimated using the Subspace-based approach in Section 2.3 or other existing order estimation methods in the literature. More details are provided in Chapter 5.*

Table 3.7: System order, N_{SFP} and N_{EC} for each system

System	order (n)	N_{SFP}	N_{EC}
I	4	2	$n+1$
II	2	9	N_{SFP}
III	4	11	N_{SFP}
IV	2	12	N_{SFP}

For System I according to Table 3.7, we have $N_{SFP} < n$. Therefore, from the criteria in (3.20) we select $N_{EC} = n + 1$. However, for Systems II, III and IV we have $N_{SFP} > n$, as a result, we consider $N_{EC} = N_{SFP}$ for these systems. For Distillation System, the overall MIMO system order can be estimated from Subspace-based approach in Section 2.3 by considering the dominant singular values of subspace predictor matrix L_w in (2.169). Figure 3.7 shows the first 20 dominant singular values of L_w for the MIMO Distillation system. Here, the system order can be estimated as $n = 1$ or $n = 2$ and in both cases $n < N_{SFP}$. As a result, from (3.20) we have $N_{EC} = N_{SFP}$ for all subsystems.

Therefore, from the technique in Table 3.2 the efficient values of control horizons are obtained for each subsystem as Table 3.8,

To verify the efficiency of proposed control horizon selection criteria in (3.20) the stability graphs of Systems I, II and III are plotted in Figures 3.8, 3.9 and 3.10 for $1 \leq N_c \leq 14$. The results show that, since $Q = I$ is small enough and $R = 100I$ is large enough, N_c has no effect on the SPC closed-loop stability.

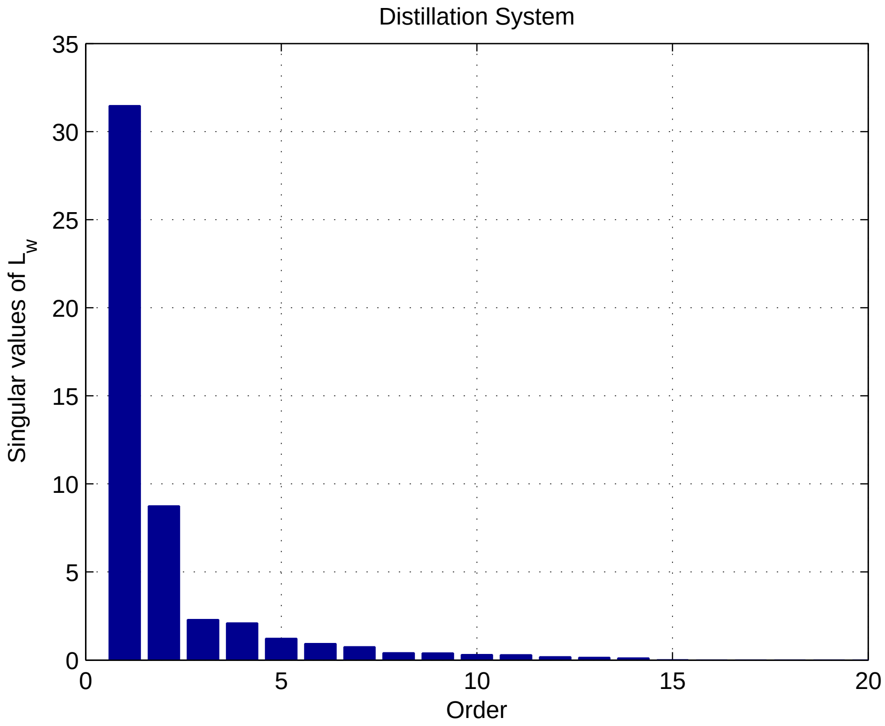

Figure 3.7: Distillation System: Singular values of L_w for Distillation System

Table 3.8: N_{EC} for each subsystem of MIMO Distillation System

Subsystem	N_{EC}
G_{p11}	3
G_{p21}	9
G_{p12}	5
G_{p22}	5

Moreover, the SPC stability graphs indicate that there is no significant change on the SPC gain, $\tilde{k}_e$, for $N_c > N_{EC}$. Therefore, the achievable performance of the SPC closed-loop system remains the same for $N_c > N_{EC}$, which verifies the proposed computationally efficient selection criteria of control horizon in (3.20).

Step 3: In this step, the N_{EP} is selected based on the SPC cost function optimization procedure in (3.2). Given $Q = I$, $R = 100I$ and $N_c = N_{EC}$ the SPC cost function in (2.117) has been evaluated for feasibility range of each system, $N_{SFP} \leq N_p \leq M$. From the evaluation result N_{EP} is selected for each system according to the Table 3.9.

From the proposed technique in Table 3.2 the efficient values of prediction horizons are obtained

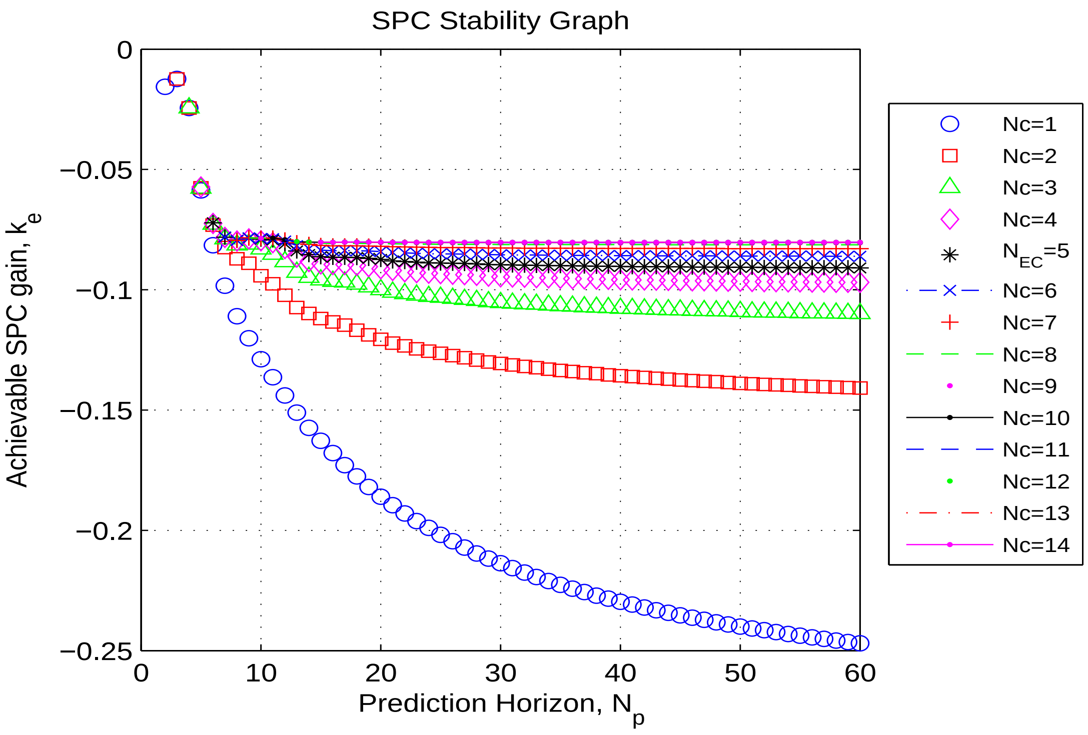

Figure 3.8: System I: SPC stability graphs for $Q = I$, $R = 100I$ and $1 \leq N_c \leq 14$

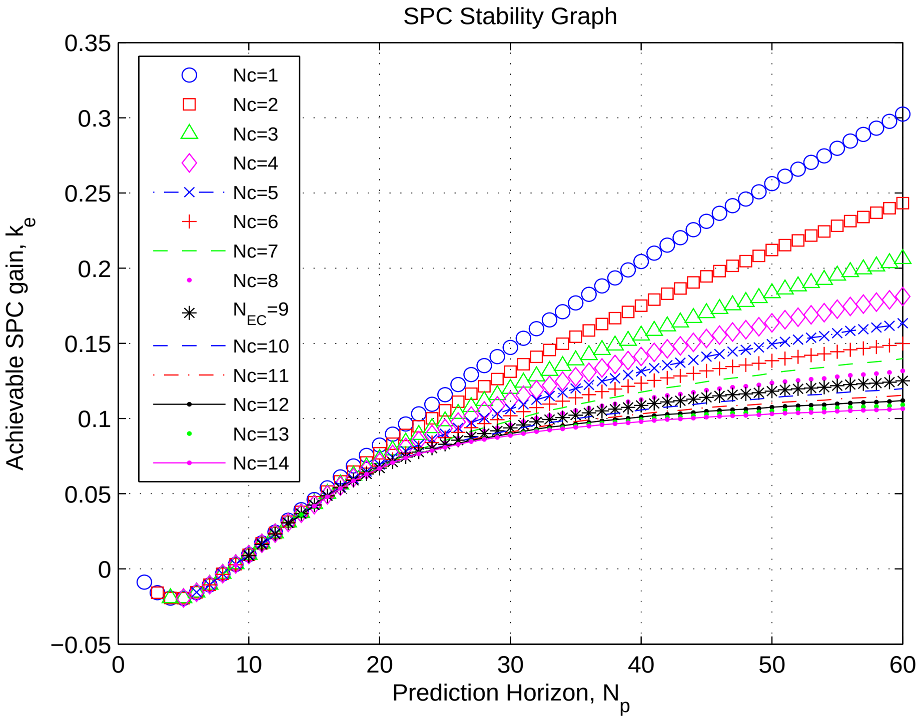

Figure 3.9: System II: SPC stability graphs for $Q = I$, $R = 100I$ and $1 \leq N_c \leq 14$

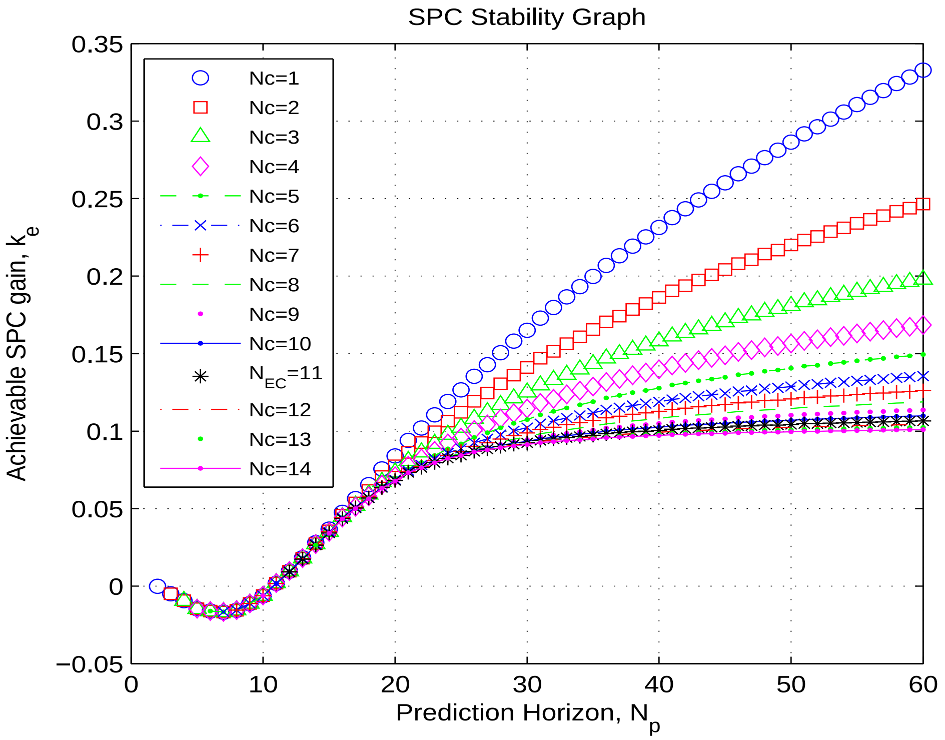

Figure 3.10: System III: SPC stability graphs for $Q = I$, $R = 100I$ and $1 \leq N_c \leq 14$

Table 3.9: Selected N_{EP} by SPC Cost Function Evaluation

System	N_{EP}
I	15
II	30
III	32
IV	50

for each subsystem of the MIMO Distillation System by minimizing the cost function of each subsystem. The results are shown in Table 3.10,

Table 3.10: N_{EP} for each subsystem of MIMO Distillation System

Subsystem	N_{EP}
G_{p11}	32
G_{p21}	30
G_{p12}	24
G_{p22}	16

Figures 3.11 and 3.12 illustrate the SPC cost function evaluation and selection of N_{EP} for Systems III and IV.

Moreover, the SPC stability graphs in Figures 3.8, 3.9 and 3.10 show that for $N_c = N_{EC}$

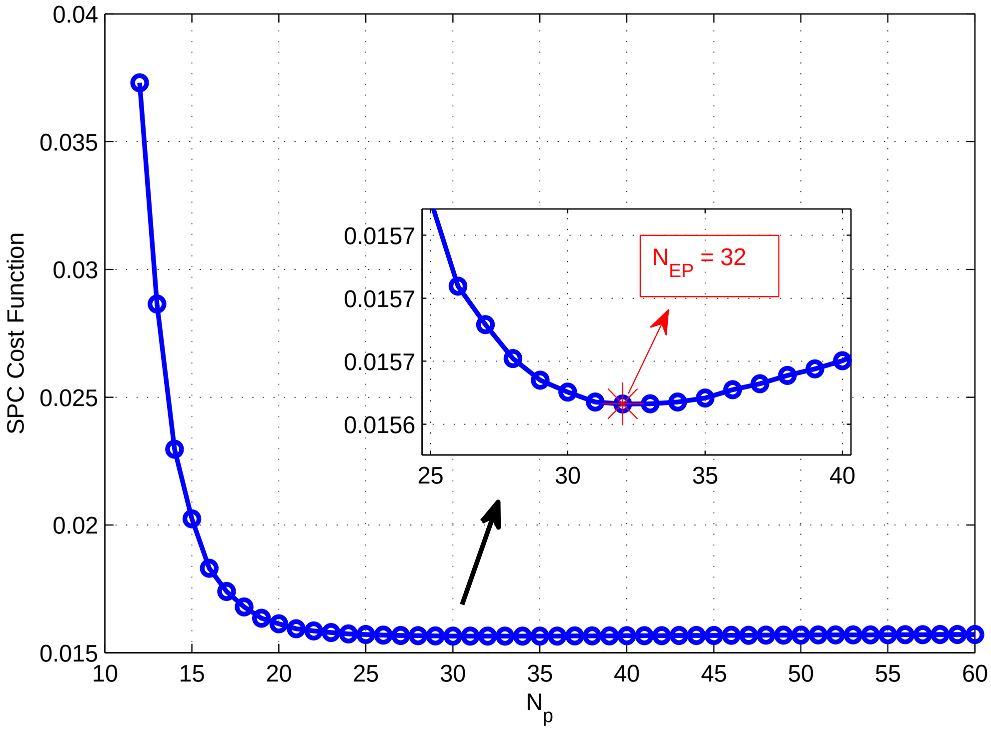

Figure 3.11: System III: SPC cost function evaluation for $Q = I$, $R = 100I$ and $N_c = N_{EC} = 11$

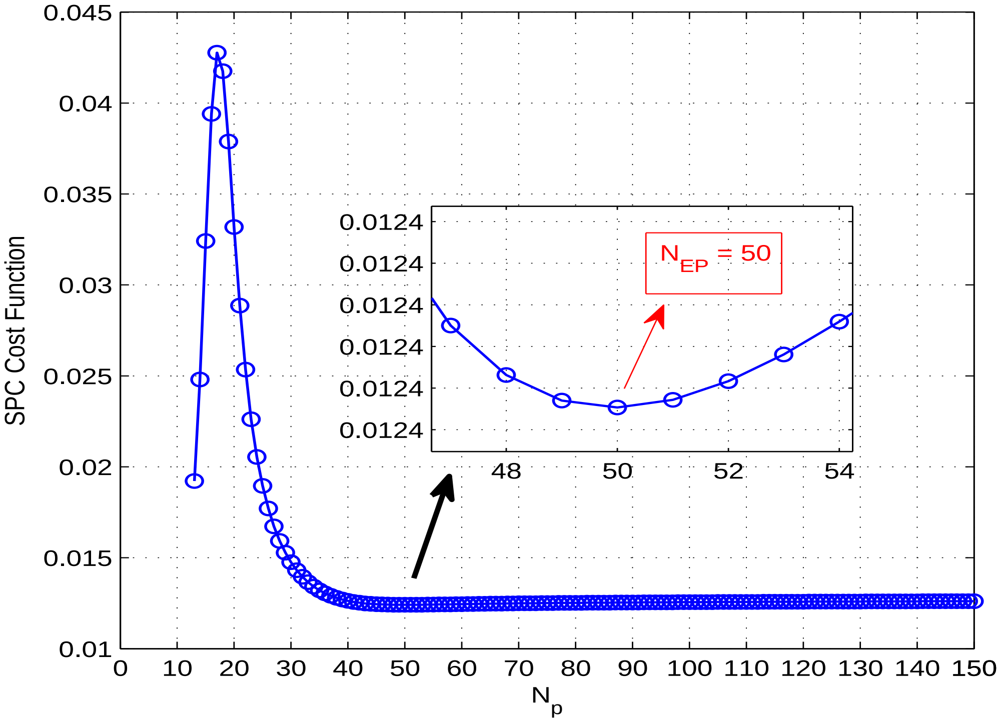

Figure 3.12: System IV: SPC cost function evaluation for $Q = I$, $R = 100I$ and $N_c = N_{EC} = 12$

increasing the prediction horizon from N_{EP} for each system has no significant effect on the SPC gain $\tilde{k}_e$. As a result, it can not enhance the maximum achievable SPC closed-loop performance, but will increase the computational load of the SPC algorithm.

3.4.4 Effect of Weighting Matrices R and Q

According to Theorem 3.2.1 and Remark 3.2.3 to ensure the SPC closed-loop stability the weighting matrices Q and R need to be selected sufficiently small and sufficiently large, respectively. Here, the effect of selecting different weighting matrices R and Q on N_{SFP} and FR of N_p is studied, and the results are provided for System III.

Effect of Weighting Matrix R

In this part effects of selecting different weighting matrices, R, on the SPC closed loop stability and robustness is being studied for System III. In order to study the effects of selecting weighting matrix, R on N_{SFP} and FR of N_p, the SPC stability graphs of System III are plotted for different R values. The results are shown in Figures 3.13 and 3.14 for $Q = I$ and $N_c = N_p - 1$ for different weighting matrices R.

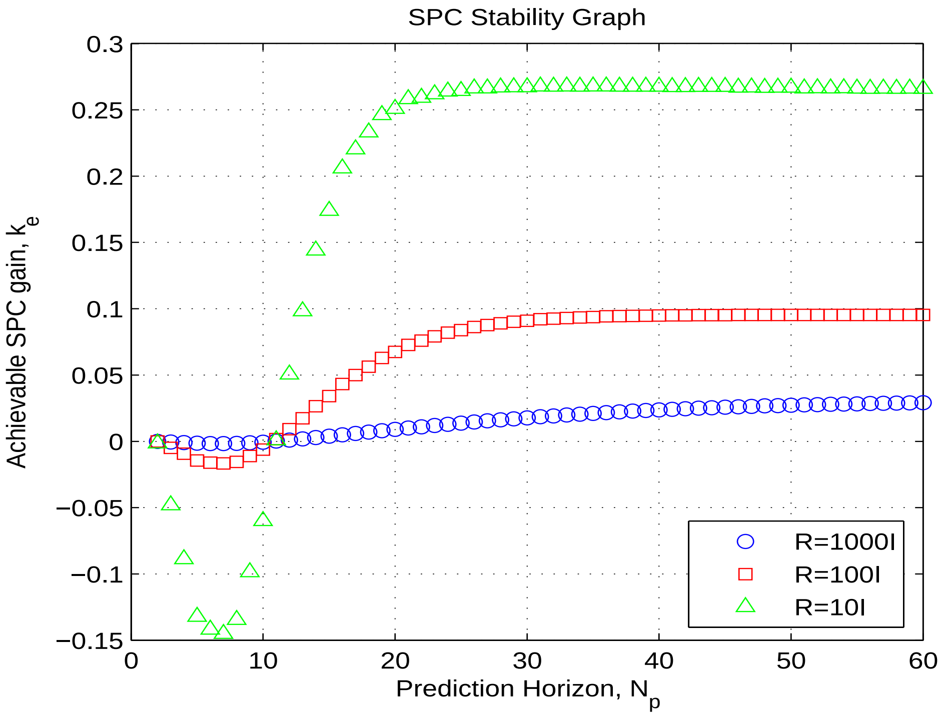

Figure 3.13: System III: SPC stability graphs for $Q = I$ and $N_c = N_p - 1$

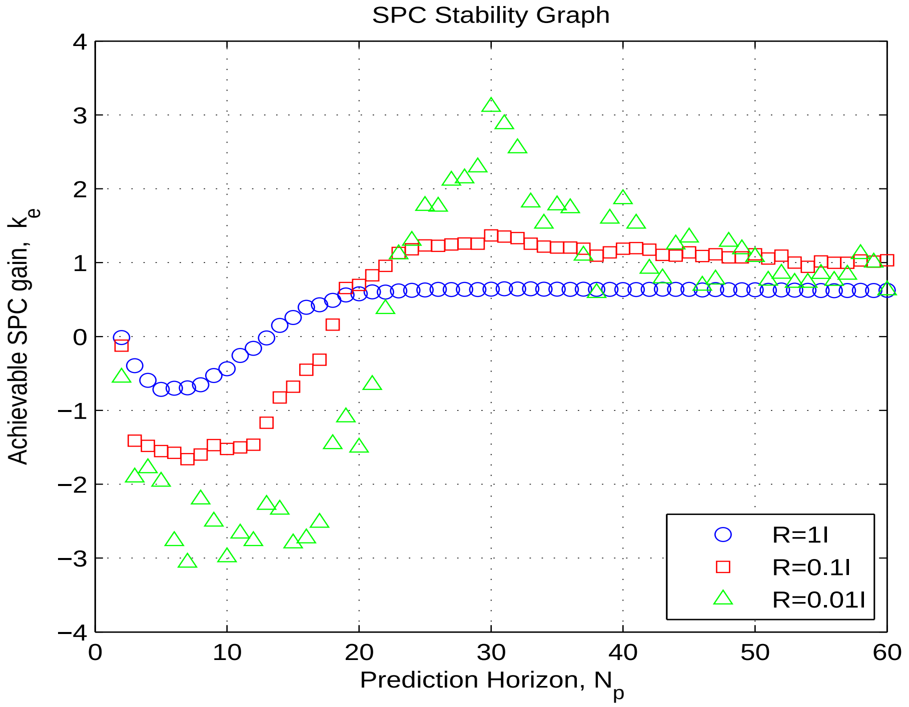

Figure 3.14: System III: SPC stability graphs for $Q = I$ and $N_c = N_p - 1$

Table 3.11 shows N_{SFP} and FR of N_p for each R in System III.

Table 3.11: System III: FR of N_p for different R, $Q = I$ and $N_c = N_p - 1$

R	N_{SFP}	FR of N_p
$1000I$	11	$11 \leq N_p \leq M$
$100I$	11	$11 \leq N_p \leq M$
$10I$	11	$11 \leq N_p \leq M$
I	14	$14 \leq N_p \leq M$
$0.1I$	18	$18 \leq N_p \leq M$
$0.01I$	22	$22 \leq N_p \leq M$

The results show that selecting very large values for R increases robustness of the closed-loop system, but does not affect the N_{SFP} and FR of N_p. However, for very small values of R, N_{SFP} increases drastically and robustness decreases. For example, Figure 3.14 shows that for $R = 0.01I$ there is a significant fluctuation in the SPC gain value, $\tilde{k}_e$, which is not acceptable even when closed-loop system is stable. As a result, robustness and stability of the closed-loop system decreases by decreasing the values of R, which needs to be selected large enough to ensure the stability and performance, however selecting very large values of R may result a slow response. Note that this behavior was expected based on our SPC stability analysis in Theorem 3.2.1 and similar results can be obtained by analyzing other systems.

Effect of Weighting Matrix Q

To investigate the effect of different weighting matrices Q on N_{SFP} and FR of N_p, the SPC stability graphs are considered for $R = 100I_{N_c \times N_c}$ and $N_c = N_p - 1$. Figures 3.15 and 3.16 show the SPC stability graphs of System III for different values of matrix Q.

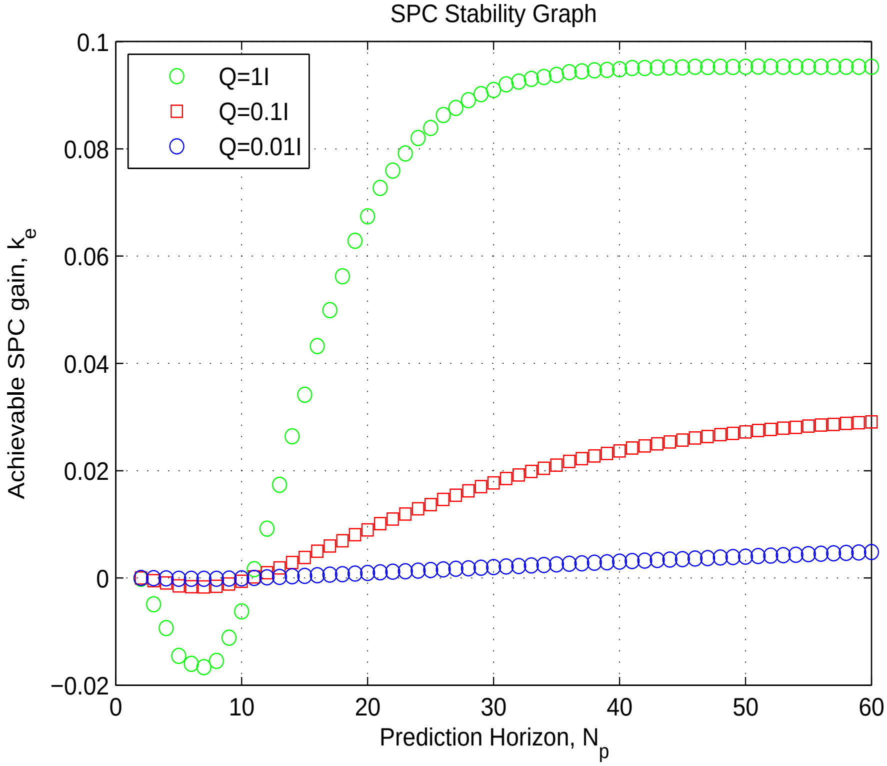

Figure 3.15: System III: SPC stability graphs for $R = 100I$ and $N_c = N_p - 1$

The results show that selecting very large values for Q decreases both robustness and FR of N_p and increases the N_{SFP}. Therefore, the closed-loop stability can be jeopardized by selecting large Q weighting matrices. On the other hand, for small value of matrix Q, robustness increases, but N_{SFP} and FR of N_p remain same. Table 3.12 shows N_{SFP} and FR of N_p for each value of matrix Q for System III.

3.4.5 Performance Evaluation

In order to verify the effectiveness of the proposed technique in Table 3.2, the designed SPC by proposed technique of SPC horizons selection is applied to System IV and MIMO Distillation System. In this simulation, the following SPC tuning parameters and constraints are selected for

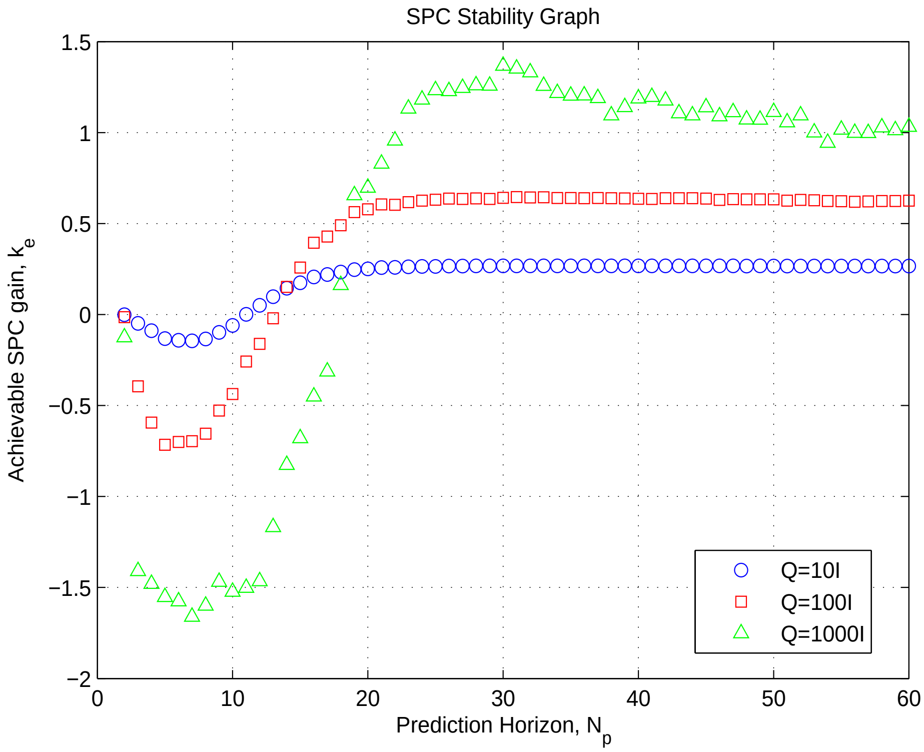

Figure 3.16: System III: SPC stability graphs for $R = 100I$ and $N_c = N_p - 1$

Table 3.12: System III: FR of N_p for different Q, $R = 100I$ and $N_c = N_p - 1$

Q	N_{SFP}	FR of N_p
$1000I$	18	$18 \leq N_p \leq M$
$100I$	14	$14 \leq N_p \leq M$
$10I$	11	$11 \leq N_p \leq M$
I	11	$11 \leq N_p \leq M$
$0.1I$	11	$11 \leq N_p \leq M$
$0.01I$	11	$11 \leq N_p \leq M$

each system,

System IV: $Q = I, \quad R = 100I, \quad N_c = N_{EC} = 12, \quad N_p = N_{EP} = 50, \quad -0.1 \leq \Delta u \leq 0.1, \quad -3 \leq u \leq 3$

System V: $Q = 0.01I, \quad R = 400I, \quad N_c = N_{EC} = 9, \quad N_p = N_{EP} = 32, \quad -0.1 \leq \Delta u \leq 0.1, \quad -1 \leq u \leq 1$

Figures 3.17, 3.18, 3.19 and 3.20 illustrate the SPC closed-loop system response of designed SPC and the control signal for System IV and MIMO Distillation System. The figures show the effectiveness of the proposed SPC design method in improving stability and performance of the both SISO and MIMO systems, and the fact that control signals remain in the prescribed constraints limit.

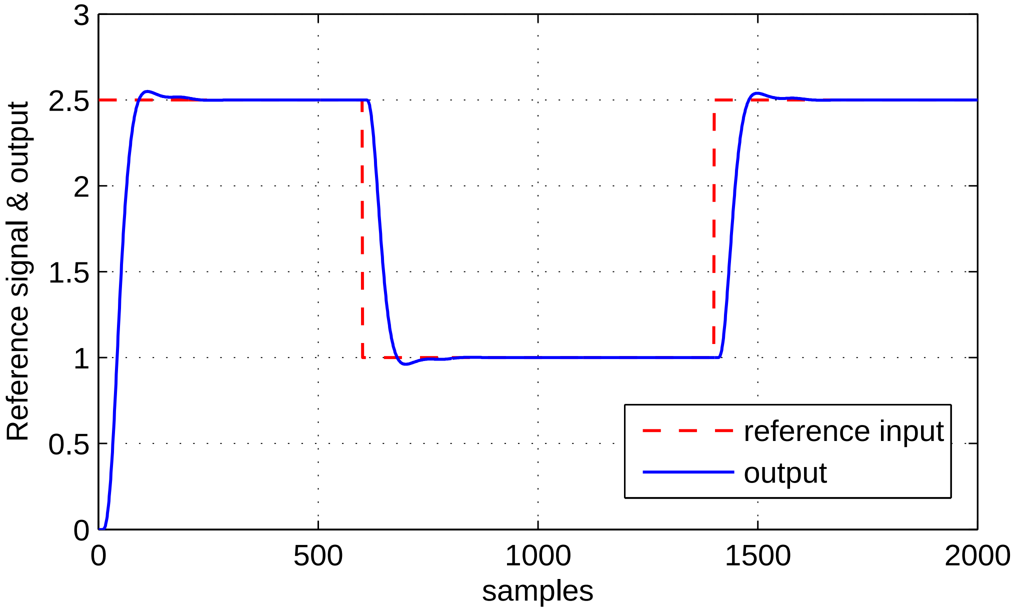

Figure 3.17: System IV: Reference input and output for $Q = I$, $R = 100I$, $N_c = N_{EC} = 12$ and $N_p = N_{EP} = 50$

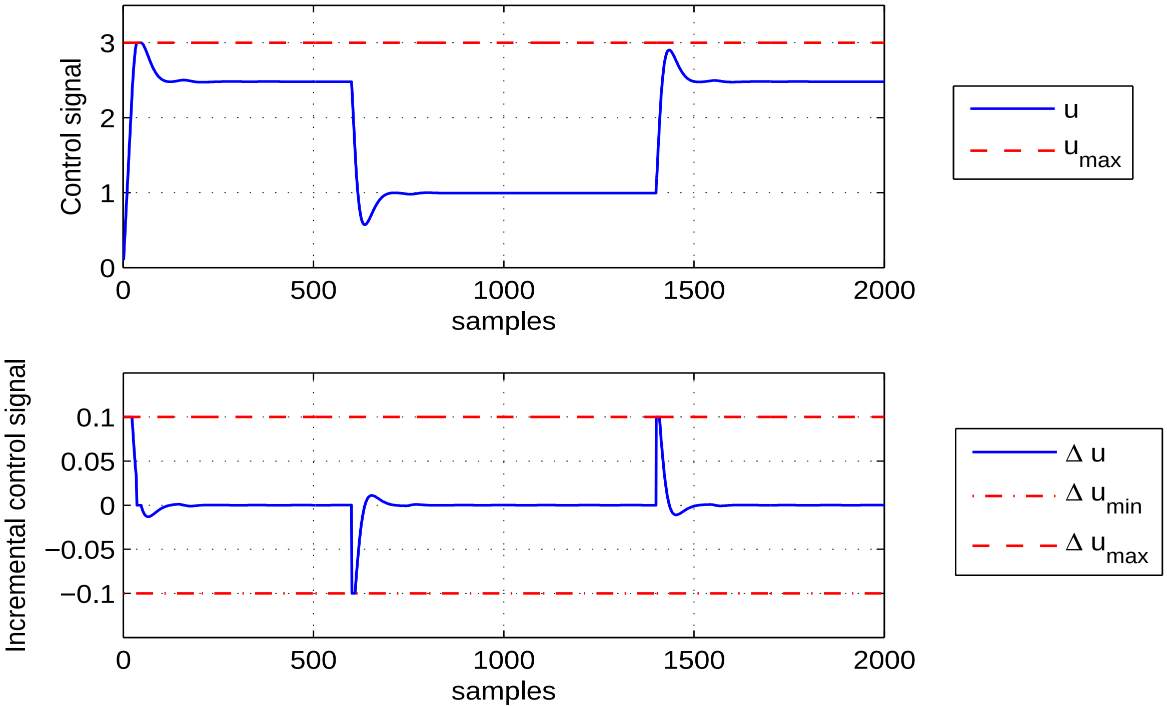

Figure 3.18: System IV: Control signal u, its variation Δu and the constraints for $Q = I$, $R = 100I$, $N_c = N_{EC} = 12$ and $N_p = N_{EP} = 50$

3.4.6 Effect of Prediction Horizon

In order to verify the effectiveness of the proposed technique in Table 3.2, the designed SPC by proposed technique of SPC horizons selection is applied to SISO Systems II and III. In this

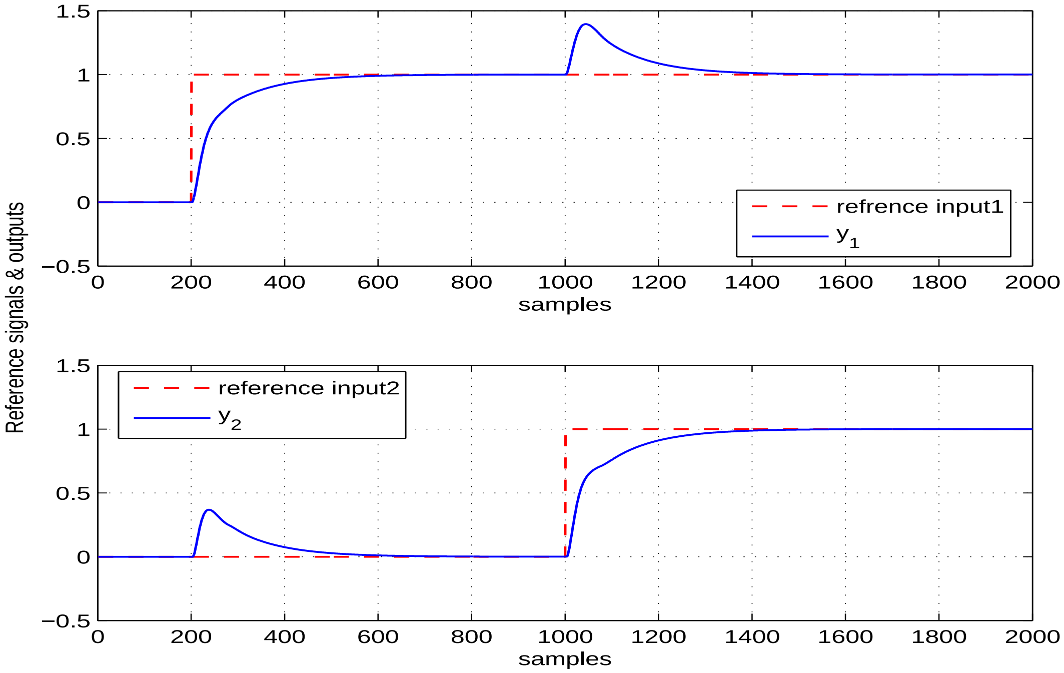

Figure 3.19: Distillation System: Reference inputs and outputs y_1 and y_2 for $Q = 0.01I$, $R = 400I$, $N_c = N_{EC} = 9$ and $N_p = N_{EP} = 32$

simulation, the following SPC tuning parameters and constraints are selected for each system,

System II: $\quad Q = I,\ \ R = 100I,\ \ N_c = N_{EC} = 9,\ \ N_p = N_{EP} = 30,\ \ -0.1 \leq \Delta u \leq 0.1,\ \ -1 \leq u \leq 1$

System III: $\quad Q = I,\ \ R = 100I,\ \ N_c = N_{EC} = 11,\ \ N_p = N_{EP} = 32,\ \ -0.1 \leq \Delta u \leq 0.1,\ \ -1 \leq u \leq 1$

Figures 3.21 3.22, 3.23 and 3.24 show the output of the SPC closed-loop system for a constant reference signal and SPC control signal and its variation for Systems II and III. Here, different values of feasible N_ps are considered to compare the performance. The figures confirm the efficiency of the proposed method in selecting the SPC horizons. According to the Figures 3.23 and 3.21 for $N_p < N_{EP}$ performance is deteriorated, but for $N_p > N_{EP}$ there is no significant changes in output of the systems. Moreover, Figures 3.22 and 3.24 show that the control signals and their variation do not violate the determined constraint values.

3.4.7 Effect of Disturbance and Noisy Data

In order to investigate performance of the proposed SPC design method in the presence of noisy data, noisy version of the System IV and MIMO Distillation System are considered as follows,

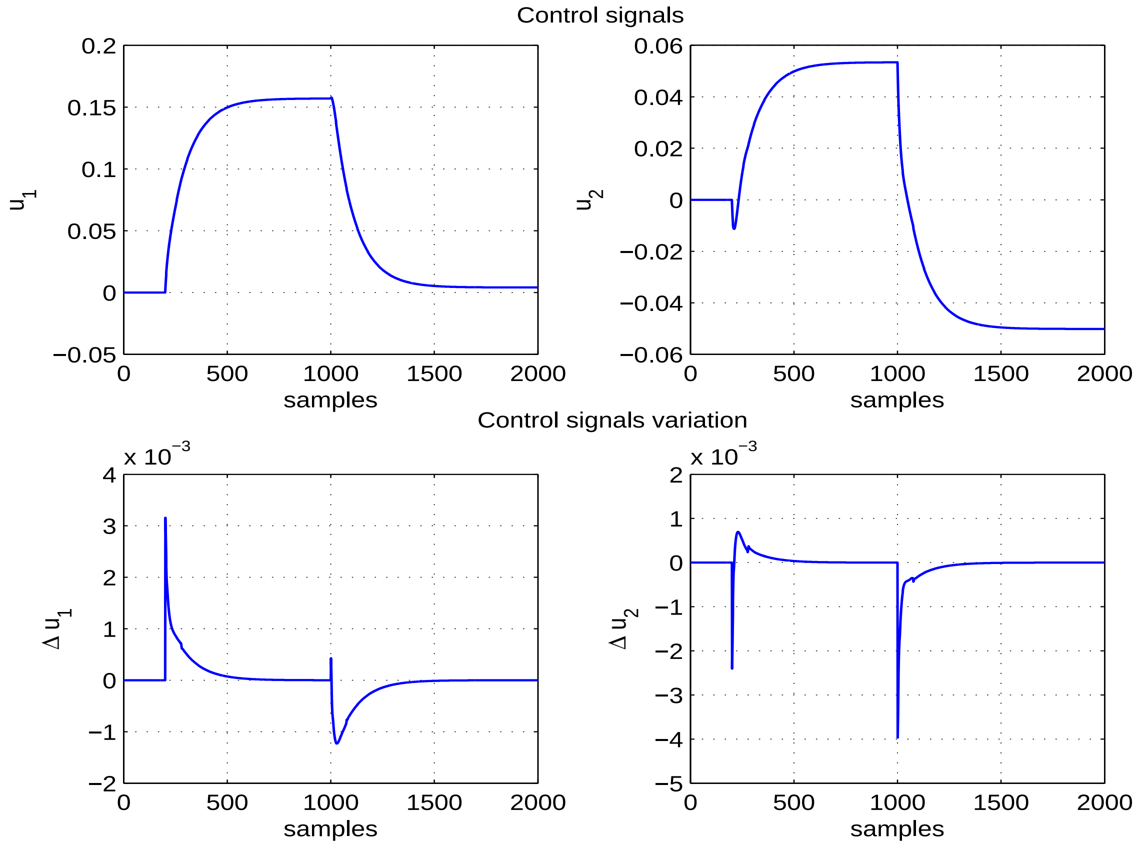

Figure 3.20: Distillation System: Control signals u_1, u_2 and their variations Δu_1, Δu_2 for $Q = 0.01I$, $R = 400I$, $N_c = N_{EC} = 9$ and $N_p = N_{EP} = 32$

System IV: SISO time-delayed noisy system

The transfer function of System IV in Section 3.4.1 with additive noise [55],

$$Y(s) = \frac{e^{-50s}}{(150s+1)(25s+1)}U(s) + E_w(s) \tag{3.30}$$

where E_w is a Gaussian noise with zero mean and variance of σ^2. Appropriate variances are selected for simulation to obtain the desired SNRs.

Figure 3.25 shows the SPC stability graphs of System IV for noisy data with $SNR = 0$db and $SNR = 10$db for $Q = I$ and $R = 100I$.

From the SPC stability graph in Figure 3.25 the N_{SFP} and FR of N_p are determined based on the SPC stability condition in (3.15) to ensure the closed-loop stability. Efficient values of control and prediction horizons are determined by applying the technique in Table 3.2. For $SNR = 0$db

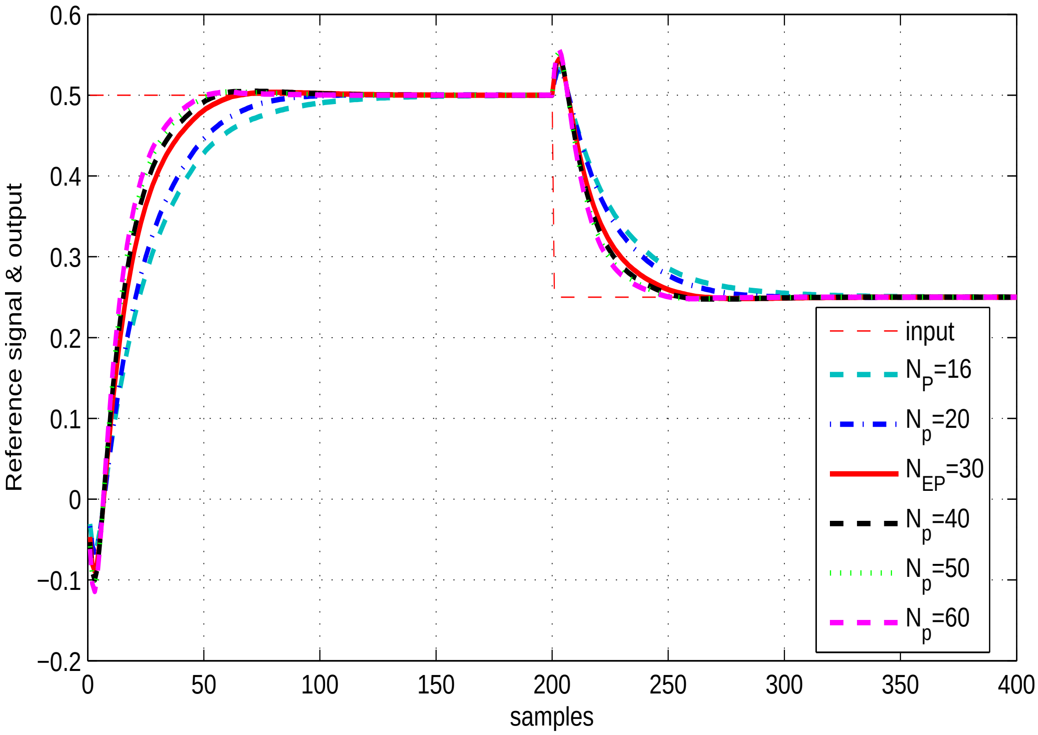

Figure 3.21: System II: Reference signal and output for $Q = I$, $R = 100I$, $N_c = N_{EC} = 9$ and different values of feasible N_p

we have $n + 1 < N_{SFP}$, then $N_{EC} = 16$, and for $SNR = 10$db we have $n + 1 > N_{SFP}$, therefore, $N_{EC} = 3$. The results are shown in Table 3.13 for each SNR.

Table 3.13: N_{SFP}, FR of N_p, N_{EC} and N_{EP} for System IV from Noisy Data ($M = 150$)

SNR	N_{SFP}	FR of N_p	N_{EC}	N_{EP}
0db	16	$16 \leq N_p \leq M$	16	35
10db	2	$2 \leq N_p \leq M$	3	37

Figures 3.26 and 3.27 show the SPC closed-loop system responses for this noisy system with the following SPC parameters and constraints for each SNR case,

$$SNR = 0db: \quad N_{EC} = 16, \quad N_{EP} = 35, \quad Q = I, \quad R = 100I, \quad -1 \leq u(k) \leq 1, \quad -0.1 \leq \Delta u \leq 0.1$$

$$SNR = 10db: \quad N_{EC} = 3, \quad N_{EP} = 37, \quad Q = I, \quad R = 100I, \quad -1 \leq u(k) \leq 1, \quad -0.1 \leq \Delta u \leq 0.1$$

The results show the effectiveness of the proposed SPC design method even in the presence of noisy data.

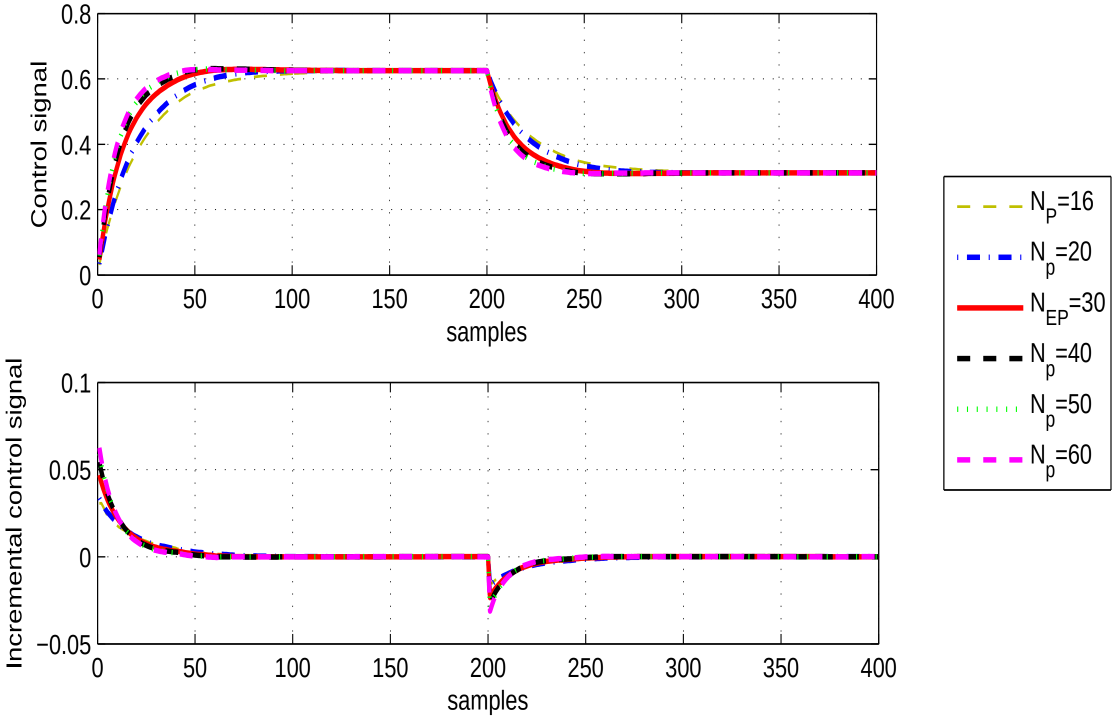

Figure 3.22: System II: SPC control signal and its variation for $Q = I$, $R = 100I$, $N_c = N_{EC} = 9$ and different values of feasible N_p

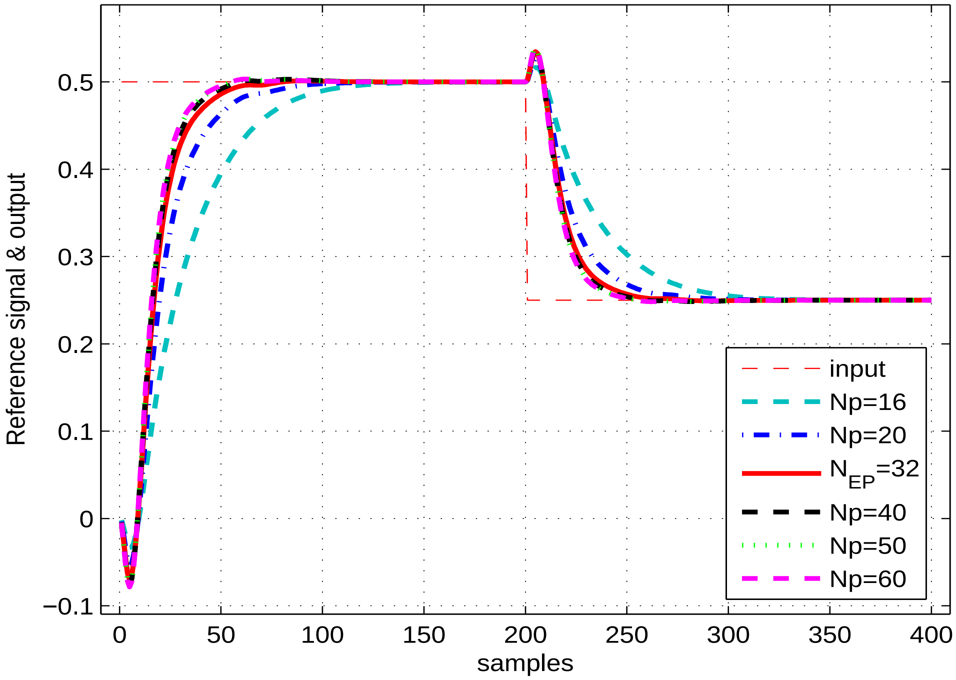

Figure 3.23: System III: Reference signal and output for $Q = I$, $R = 100I$, $N_c = N_{EC} = 11$ and different values of feasible N_p

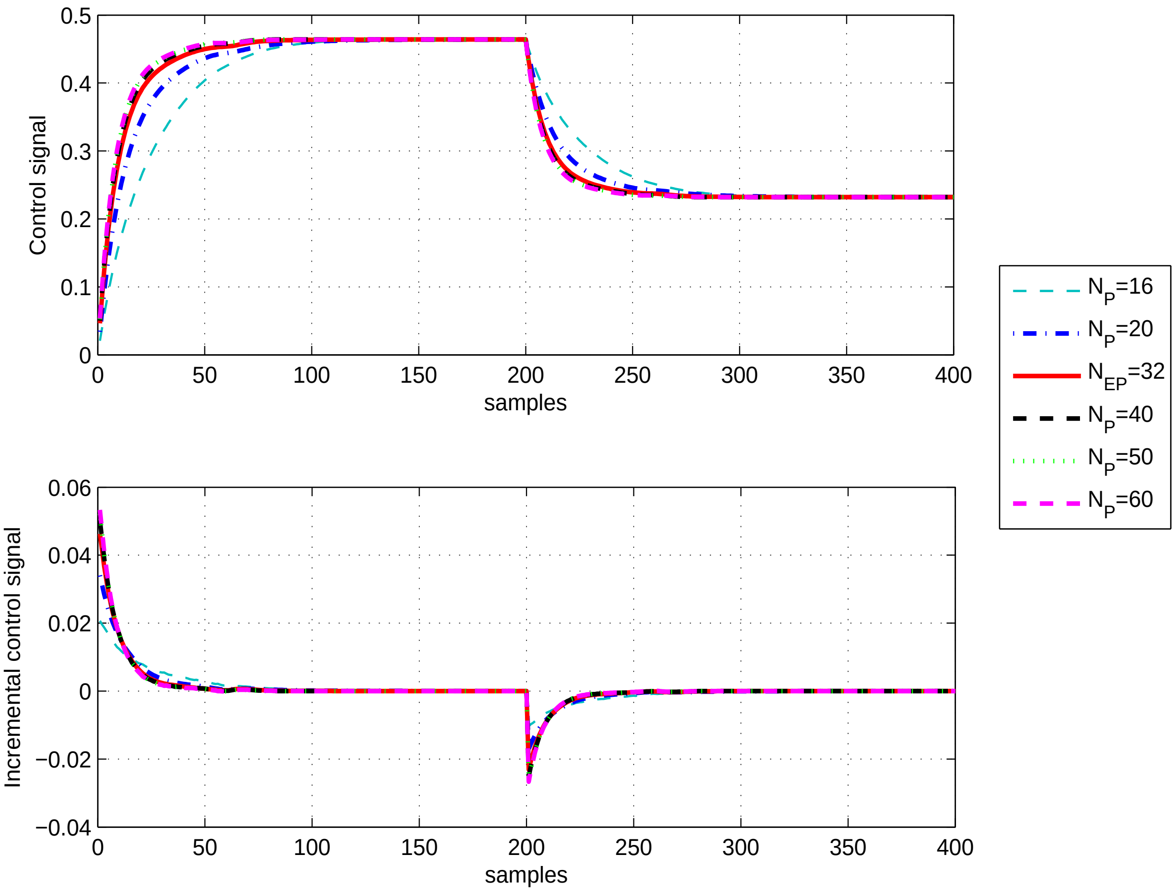

Figure 3.24: System III: SPC control signal and its variation for $Q = I$, $R = 100I$, $N_c = N_{EC} = 11$ and different values of feasible N_p

Distillation System: MIMO time-delayed system

Disturbance added Distillation System model in Section 3.4.1 [53],

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{16.7s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} + \begin{bmatrix} \frac{3.8e^{-8s}}{10.9s+1} \\ \frac{4.9e^{-3s}}{13.2s+1} \end{bmatrix} E_d(s) \tag{3.31}$$

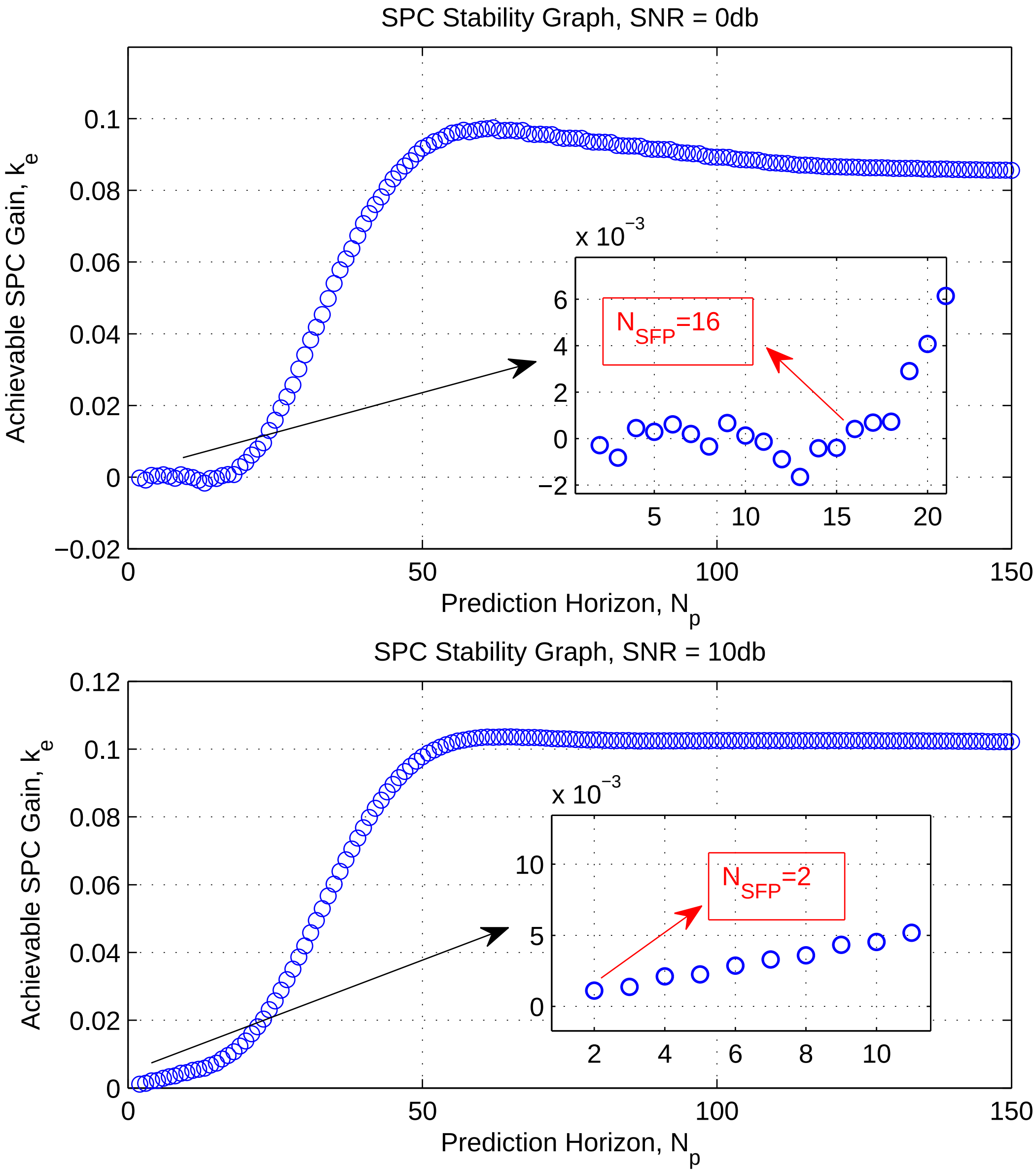

Figure 3.25: System IV: SPC stability graphs from noisy I/O data for SNR=0db and SNR=10db

where E_d is an independent white noise with covariance of $diag\{4.4582,\ 4.9898\}$. The discretized disturbance model using zero order holder with $T_s = 1min$ is obtained as,

$$G_d(z^{-1}) = \begin{bmatrix} z^{-9}\frac{0.331}{1-0.9123z^{-1}} \\ z^{-4}\frac{0.3575}{1-0.927z^{-1}} \end{bmatrix} \tag{3.32}$$

To determine the FR of prediction horizon, which guarantees the SPC closed-loop stability,

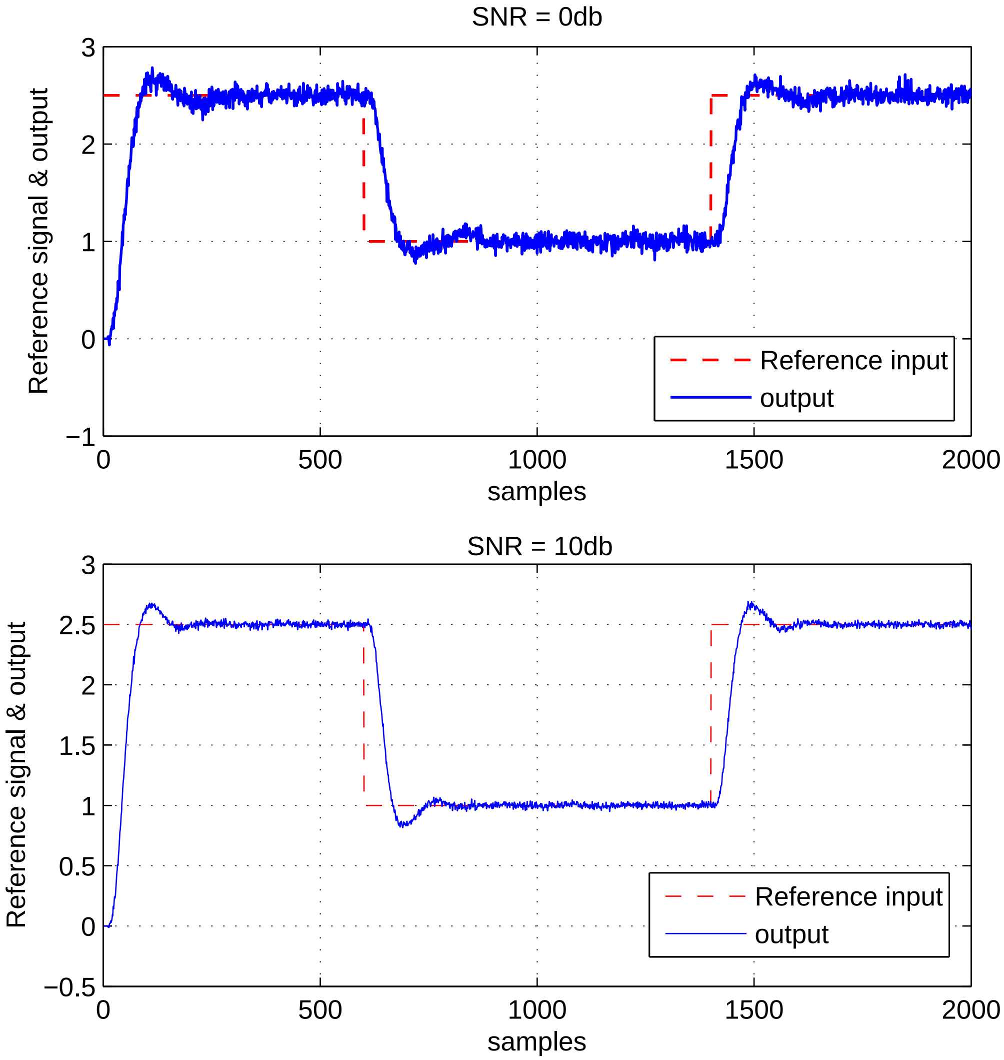

Figure 3.26: System IV: Reference input and output for noisy system for SNR=0db and SNR=10db

the SPC stability graphs need to be plotted for each subsystem. Figure 3.28 shows the SPC stability graphs from I/O data of System V with disturbance for each subsystem with $R = 400I$ and $Q = 0.01I$.

The N_{SFP} and FR of N_p are determined based on the SPC stability condition in (3.15) by using the SPC stability graphs to ensure the closed-loop stability. The results are shown in Table 3.14. Note that from Figure (3.28) it can be seen that in subsystem G_{p12} for $N_p > 66$ the SPC gain k_e becomes positive. Therefore, FR of N_p for subsystem G_{p12} will be $8 \leq N_p \leq 66$.

Therefore, by applying the proposed technique in Table 3.2 the efficient values of control and

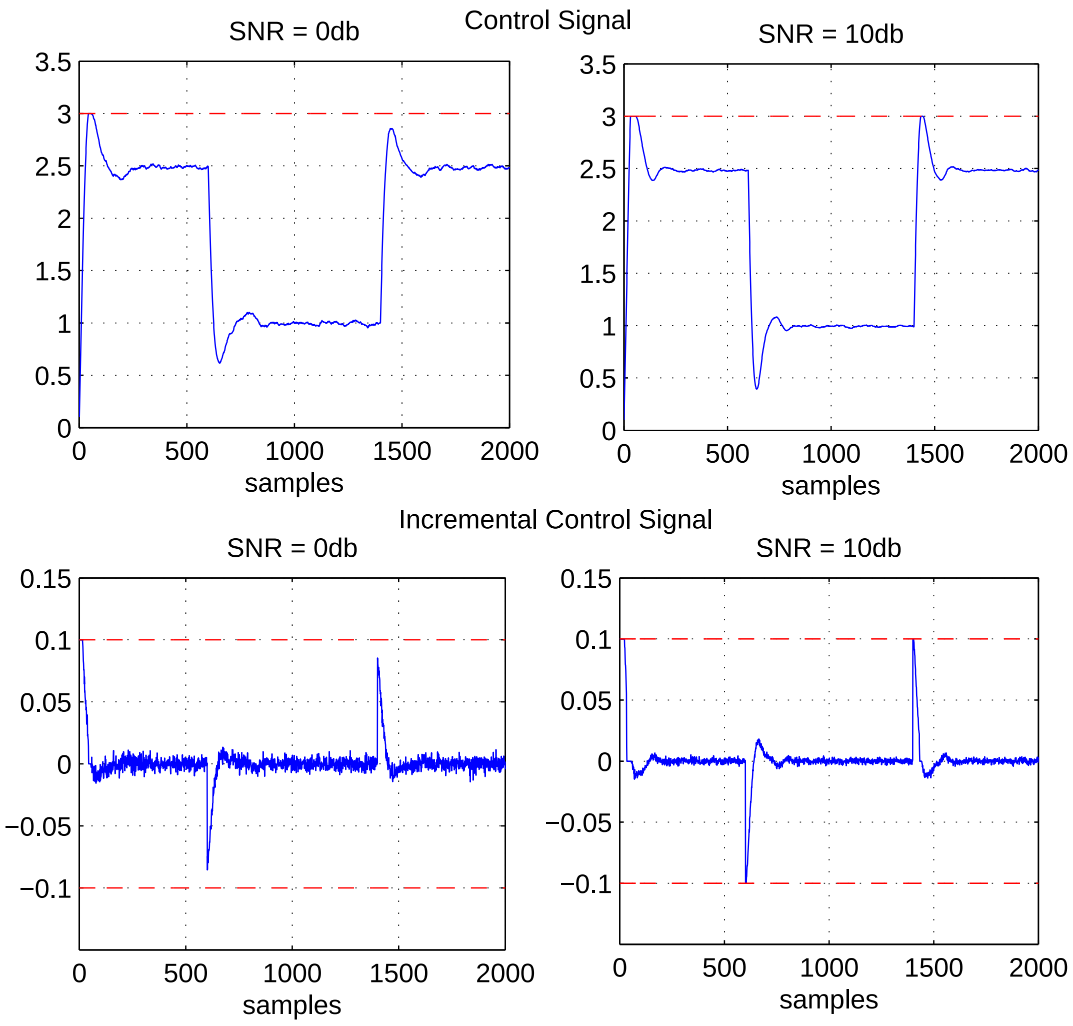

Figure 3.27: System IV: Control signal u and its variations Δu for noisy system for SNR=0db and SNR=10db

Table 3.14: N_{SFP} and FR of N_p for Distillation System with Disturbance $(M = 80)$

Subsystem	N_{SFP}	FR of N_p
G_{p11}	5	$5 \leq N_p \leq M$
G_{p21}	2	$2 \leq N_p \leq M$
G_{p12}	8	$8 \leq N_p \leq 66$
G_{p22}	2	$2 \leq N_p \leq M$

prediction horizons are determined as $N_{EC} = 8$ and $N_{EP} = 30$. Figures 3.29 and 3.30 show the SPC closed-loop system responses for this noisy MIMO system with the following SPC parameters

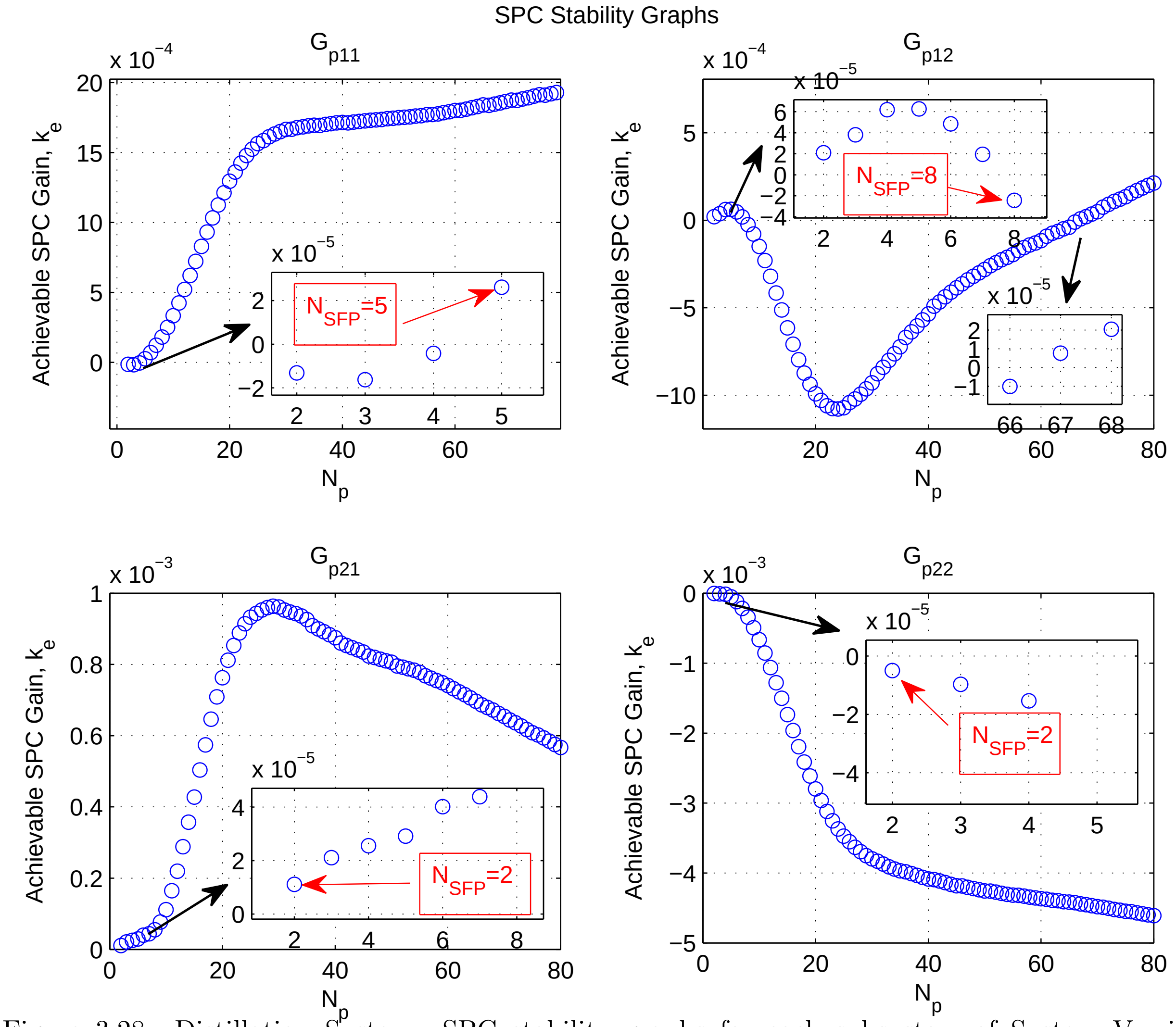

Figure 3.28: Distillation System: SPC stability graphs for each subsystem of System V with disturbance

and constraints,

$$N_{EC} = 8, \quad N_{EP} = 30, \quad Q = 0.01I, \quad R = 400I, \quad -1 \leq u(k) \leq 1, \quad -0.1 \leq \Delta u \leq 0.1$$

Remark 3.4.3. *Comparing the results of noise-free and noisy scenario shows that different SPC stability graphs and different FR of N_p are obtained in those scenarios. This observation proves generality and effectiveness of the proposed data-driven SPC stability analysis.*

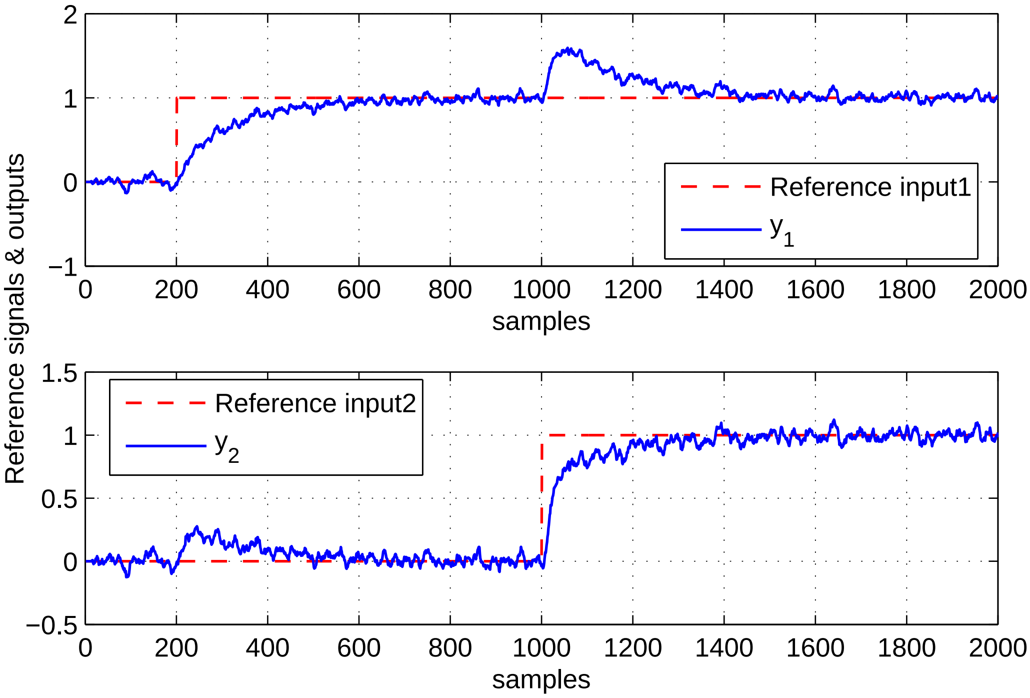

Figure 3.29: Distillation System: Reference inputs and outputs y_1 and y_2 for System V with disturbance

3.5 Summary

In this chapter the necessary and sufficient condition to assure the SPC closed-loop stability was provided for open-loop stable systems. Based on the derived SPC stability condition, a model-free technique (Table 3.1) was provided to determine the shortest-feasible-prediction-horizon (N_{SFP}) and the feasibility range (FR) of prediction horizon that guarantees the SPC closed-loop stability. Moreover, the criteria to optimally determine the efficient control horizon (N_{EC}) and efficient prediction horizon (N_{EP}) were suggested based on efficiently minimizing the dimension of the subspace predictor and optimizing the SPC cost function. Consequently, the technique in Table 3.2 was proposed for tuning SPC horizons based on the technique in Table 3.1, and provided N_{EC} and N_{EP}. The proposed techniques were evaluated in simulation on three SISO non-minimum phase stable systems, a SISO time-delayed noisy system and a MIMO time-delayed system with disturbance. For each of the systems our stability criteria successfully provided the necessary and sufficient FR of N_p and N_{SFP}. Consequently, applying the proposed model-free technique on each system determined the efficient SPC horizons among the feasibility range by minimizing the desired SPC cost function as well. The simulation results show effectiveness of the proposed SPC design

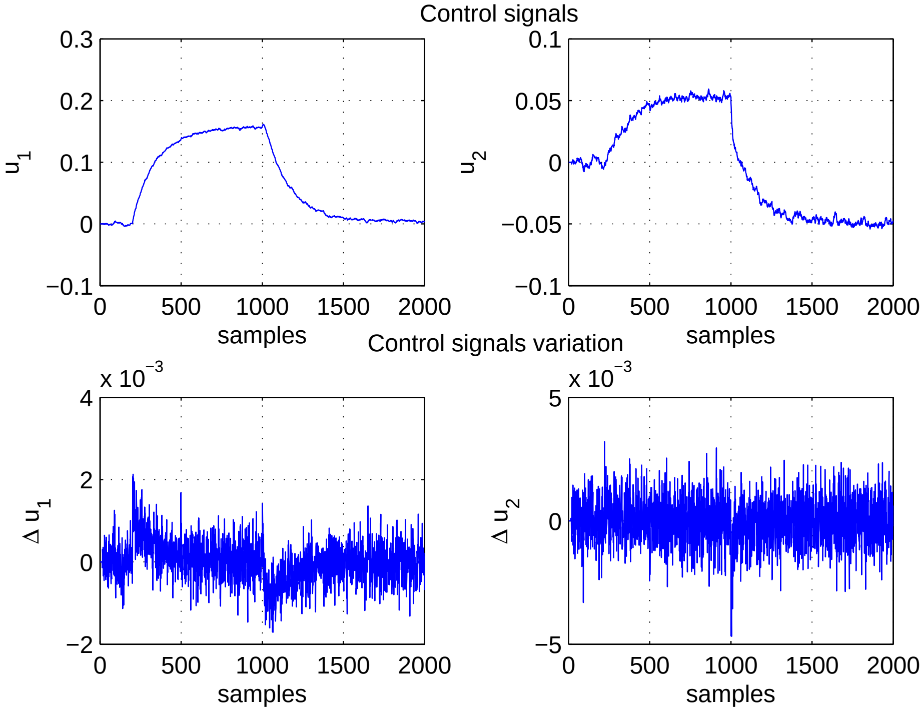

Figure 3.30: Distillation System: Control signals u_1, u_2 and their variations Δu_1, Δu_2 for System V with disturbance

method.

Chapter 4

SPC Gains Updating: PSO-based FGS-SPC

4.1 Problem Statement and Chapter Summary

Main advantage of SPC is its capability to on-line adapting the SPC gains directly from I/O data, which makes SPC appropriate to control time-varying and nonlinear systems. In this SPC gain updating procedure, the subspace linear predictor matrices, L_u and L_w, and subsequently, the controller gain matrices, $\tilde{K}_e$ and $\tilde{K}_{\Delta w_p}$, must be calculated off-line by using equations (2.133), (2.148) and (2.149) before the controller implementation. In order to adapt the controller with any parameter uncertainties in the time-varying or nonlinear systems, the SPC matrices must be updated by applying randomly generated persistent excitation (PE) signals to the system and capturing the new I/O data from the system during the closed-loop control. After that, to obtain the new incremental control law from equation (2.150) the controller gain matrices, $\tilde{K}_e$ and $\tilde{K}_{\Delta w_p}$, have to be recalculated by using the equations (2.148), (2.149) and by considering the new updated L_u and L_w matrices. However, this procedure is very time consuming and demands high computational load, which increases by system dimension [68]. On the other hand, applying the PE input signals to the system can interrupt the system operation at the steady-state mode and may cause a biased estimate when applying to the closed-loop system [100, 134]. It will be advantageous for SPC to update the controller gains $\tilde{K}_e$ and $\tilde{K}_{\Delta w_p}$ without the mentioned drawbacks.

In this chapter, a Particle Swarm Optimization-based Fuzzy Gain-Scheduling (PSO-based FGS) method is proposed to update the SPC gains for time-varying systems in the presence of constraints and noise. The method is denoted by PSO-based FGS-SPC. In the proposed method, instead of updating the predictor matrices by applying the PE signals, only SPC gains are re-tunned by applying PSO-based FGS technique. Consequently, the decrease in computational load makes the proposed method more suitable and time efficient for industrial applications. Moreover, the proposed PSO-based FGS technique overcomes the problem of controller gain ranges (CGRs) calculation in FGS algorithm. In PSO-based FGS-SPC the CRGs are determined optimally by minimizing the SPC cost function.

4.2 Re-arranging SPC Gain Formula

Considering the incremental control law for unconstrained SPC in equation (2.150), this equation can be rewritten as below [135],

$$\begin{aligned}\Delta u(k+1) &= -\tilde{k}_e(y(k) - r(k+1)) - \tilde{k}_{\Delta w_p} \begin{bmatrix} \Delta y_p \\ \Delta u_p \end{bmatrix} && (4.1)\\ &= -\tilde{k}_e e(k) - [\tilde{k}_{\Delta y_p} \;\; \tilde{k}_{\Delta u_p}] \begin{bmatrix} \Delta y_p \\ \Delta u_p \end{bmatrix} && (4.2)\end{aligned}$$

where $e(k)$ is the *tracking error* between the piecewise constant reference input and the output, and $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$ are defined as [135],

$$\begin{aligned}\tilde{k}_{\Delta y_p} &= \tilde{k}_{\Delta w_p}(:, 1 : Ml) && (4.3)\\ \tilde{k}_{\Delta u_p} &= \tilde{k}_{\Delta w_p}(:, Ml + 1 : M(m + l)). && (4.4)\end{aligned}$$

and $\tilde{k}_{\Delta w_p}$ has been defined in Section 2.2.2 as the first m rows of the matrix $\tilde{K}_{\Delta w_p}$ in (2.149). Here, by considering a piecewise constant reference input, variation of the tracking error, $\Delta e(k)$, can be obtain as follows,

$$\begin{aligned}\Delta e(k) &= e(k) - e(k-1) && (4.5)\\ &= (y(k) - r(k+1)) - (y(k-1) - r(k)) && (4.6)\\ &= y(k) - y(k-1) && (4.7)\\ &= \Delta y(k) && (4.8)\end{aligned}$$

Then, the vector of incremental past I/O data, Δw_p in (2.143) can be modified as 4.9. This rearranging of the standard SPC control signal formula enables us to illustrate flexibility of SPC technique in automatically updating the controller gains [135].

$$\Delta w_p = \begin{bmatrix} \Delta y_p \\ \Delta u_p \end{bmatrix} = \begin{bmatrix} \Delta y(k-M+1) \\ \vdots \\ \Delta y(k-1) \\ \Delta y(k) \\ ---- \\ \Delta u(k-M+1) \\ \vdots \\ \Delta u(k-1) \\ \Delta u(k) \end{bmatrix} = \begin{bmatrix} \Delta e(k-M+1) \\ \vdots \\ \Delta e(k-1) \\ \Delta e(k) \\ ---- \\ \Delta u(k-M+1) \\ \vdots \\ \Delta u(k-1) \\ \Delta u(k) \end{bmatrix} = \begin{bmatrix} \Delta e_p \\ \Delta u_p \end{bmatrix} \tag{4.9}$$

Here, the control law in equation (2.151) is obtained when the constraints are not active, but it can be applied to the constrained SPC by using the formulation in Section 2.2.2.

By considering the incremental control law in (4.2) and (4.9), it can be seen that K_e is the proportional gain of the tracking error, $e(k)$, and $\tilde{k}_{\Delta y_p}$ is the proportional gain of its variations, Δe_p. Therefore, this part of the controller has similar structure as the Proportional-Derivative (PD) controller.

Remark 4.2.1. *Note that, the vector Δe_p is not only the variation of error at the current time instant, but also it consists of the variation error for past output data. This feature makes the controller much more sophisticated than a simple PD control algorithm.*

Motivated by this observation the controller gains $\tilde{k}_e$ and $\tilde{k}_{\Delta y_p}$ can be easily updated in real-time by applying one of the existing advanced model-free GS techniques [136–139] with no need to apply PE signals and update the subspace linear predictor matrices. Here, a *Particle Swarm Optimization-Based Fuzzy Gain-Scheduling* (PSO-based FGS) method is proposed to update the SPC gains $\tilde{k}_e$ and $\tilde{k}_{\Delta y_p}$. The method is denoted by PSO-based FGS-SPC.

Remark 4.2.2. *The SPC gain, $\tilde{k}_{\Delta u_p}$, in (4.2), is the proportional gain of current and past variations of controller signal, Δu_p, in closed-loop system. Therefore, the gain $\tilde{k}_{\Delta u_p}$ can also be updated according to $e(k)$ and Δw_p at each time instant by considering the aforementioned PSO-based FGS-SPC algorithm.*

4.3 PSO-based FGS-SPC

Updating the SPC controller gains $\tilde{k}_e$ and $\tilde{k}_{\Delta w_p}$ in (2.150) by applying *Particle Swarm Optimization-Based Fuzzy Gain-Scheduling* (PSO-based FGS) method is proposed in this section. Unlike the existing SPC methods, the proposed algorithm does not need applying PE signals to update the subspace linear predictor matrices, L_u and L_w. Instead, it re-calculates the controller gains $\tilde{k}_e$ and $\tilde{k}_{\Delta w_p}$ based on the PSO-based FGS technique. The proposed technique can optimally calculate the controller gain ranges (CGRs) for FGS-SPC by minimizing the SPC cost function via the PSO algorithm. As a result, the PSO-based FGS-SPC can eliminate the time consuming and disruptive procedure of applying PE signals to the system, and updating the SPC gains optimally by improving the time efficiency. These features of PSO-based FGS-SPC method make it a suitable approach for industrial applications.

4.3.1 FGS-SPC Procedure

In the proposed PSO-based FGS-SPC method, the SPC gains $\tilde{k}_e$, $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$ can be updated real-time by applying a fuzzy-logic based GS algorithm, [117], [112]. In this scheme, it is assumed that the controller gains are in the following prescribed ranges [135],

$$\begin{aligned}
\tilde{k}_e &\in [\tilde{k}_{e,min}, \quad \tilde{k}_{e,max}] && (4.10)\\
\tilde{k}_{\Delta y_p} &\in [\tilde{k}_{\Delta y_p,min}, \quad \tilde{k}_{\Delta y_p,max}] && (4.11)\\
\tilde{k}_{\Delta u_p} &\in [\tilde{k}_{\Delta u_p,min}, \quad \tilde{k}_{\Delta u_p,max}] && (4.12)
\end{aligned}$$

where $\tilde{k}_{e,min}, \tilde{k}_{e,max}, \tilde{k}_{\Delta y_p,min}, \tilde{k}_{\Delta y_p,max}, \tilde{k}_{\Delta u_p,min}$ and $\tilde{k}_{\Delta u_p,max}$ are called the controller gain ranges (CGRSs). In FGS-SPC the CGRs are vectors with compatible dimensions as the corresponding gains.

For convenience, the SPC gains, $\tilde{k}_e$, $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$, are *normalized* into the range between zero and one by considering the following linear transformations,

$$\begin{aligned}
\bar{k}_e &= \frac{\tilde{k}_e - \tilde{k}_{e,min}}{\tilde{k}_{e,max} - \tilde{k}_{e,min}} && (4.13)\\
\bar{k}_{\Delta y_p} &= \frac{\tilde{k}_{\Delta y_p} - \tilde{k}_{\Delta y_{p,min}}}{\tilde{k}_{\Delta y_{p,max}} - \tilde{k}_{\Delta y_{p,min}}} && (4.14)\\
\bar{k}_{\Delta u_p} &= \frac{\tilde{k}_{\Delta u_p} - \tilde{k}_{\Delta u_{p,min}}}{\tilde{k}_{\Delta u_{p,max}} - \tilde{k}_{\Delta u_{p,min}}} && (4.15)
\end{aligned}$$

where $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$, are the normalized SPC gains.

In FGS-SPC strategy, the normalized SPC gains are determined based on the tracking error, $e(k)$, and the variation of the past I/O data vector, Δw_p, by a set of rules of the form,

$$\textit{if} \quad e(k) \textit{ is } AA_i \quad \textit{and} \quad \Delta w_p(j) \textit{ is } BB_i, \quad \textit{then} \quad \bar{k}_e \textit{ is } CC_i, \quad \bar{k}_{\Delta y_p} \textit{ is } DD_i \quad \textit{and} \quad \bar{k}_{\Delta u_p} \textit{ is } EE_i$$

where, AA_i, BB_i, CC_i, DD_i and EE_i are *Fuzzy Subsets* for input and output variables. The fuzzy subsets are also called *Membership Functions*, and the *Membership Degree* is denoted by $\mu_{AA_i}(X)$ for each variable like X.

In general, FGS procedure consists of three main parts: *Fuzzifier*, *FGS Rules* and *Defuzzifier* as shown in Figure 2.7. First, the crisp input variables are converted to fuzzy numbers in fuzzifier, then they are used to determine updated controller gains by considering GS rules. Finally, the resultant fuzzy numbers representing the controller gains are converted to crisp values.

Fuzzification

The procedure of converting crisp input variables to fuzzy numbers is called *fuzzification.* Here, the fuzzy subsets are in *triangular* shape with seven different fuzzy subsets representing linguistic fuzzy variables *negative big* (NB), *negative medium* (NM), *negative small* (NS), *zero* (ZZ), *positive big* (PB), *positive medium* (PM), and *positive small* (PS). The shape of the fuzzy subsets are all the same for $e(k)$, Δy_p, and Δu_p. However, the minimum and the maximum ranges are different for them, and selected according to the process characteristics [135, 140]. The fuzzy subsets of input variables, $e(k)$, Δy_p and Δu_p are shown in Figure 4.1.

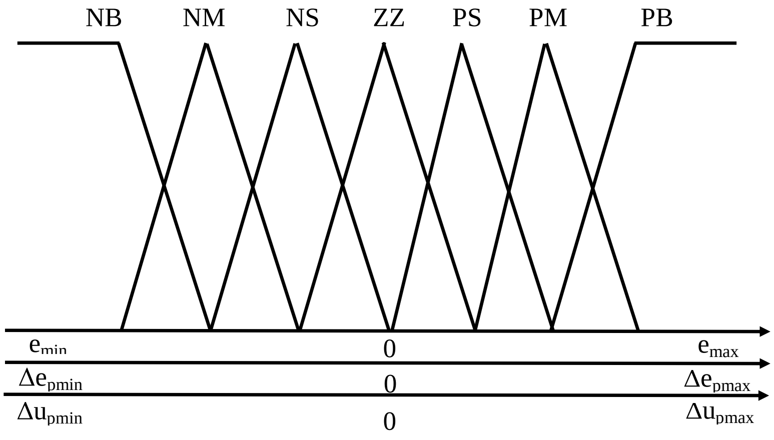

Figure 4.1: Fuzzy subsets for $e(k)$, Δe_p and Δu_p

These subsets for each input variable like X can be represented as below,

$$\mu_{NB}(X) = \begin{cases} 1 & \text{if} \quad X \leq X_{min} \\ \frac{3}{X_{min}}X - 2 & \text{if} \quad X_{min} < X \leq \frac{2}{3}X_{min} \\ 0 & \text{if} \quad otherwise \end{cases} \tag{4.16}$$

$$\mu_{NM}(X) = \begin{cases} \frac{-3}{X_{min}}X + 3 & \text{if} \quad X_{min} < X \leq \frac{2}{3}X_{min} \\ \frac{3}{X_{min}}X - 1 & \text{if} \quad \frac{2}{3}X_{min} < X \leq \frac{1}{3}X_{min} \\ 0 & \text{if} \quad otherwise \end{cases} \tag{4.17}$$

$$\mu_{NS}(X) = \begin{cases} \frac{-3}{X_{min}}X + 2 & \text{if} \quad \frac{2}{3}X_{min} < X \leq \frac{1}{3}X_{min} \\ \frac{3}{X_{min}}X & \text{if} \quad \frac{1}{3}X_{min} < X \leq 0 \\ 0 & \text{if} \quad otherwise \end{cases} \tag{4.18}$$

$$\mu_{ZZ}(X) = \begin{cases} \frac{-3}{X_{min}}X + 1 & \text{if} \quad \frac{1}{3}X_{min} < X \leq 0 \\ \frac{-3}{X_{max}}X + 1 & \text{if} \quad 0 < X \leq \frac{1}{3}X_{max} \\ 0 & \text{if} \quad otherwise \end{cases} \tag{4.19}$$

$$\mu_{PS}(X) = \begin{cases} \frac{3}{X_{max}}X & \text{if} \quad 0 < X \leq \frac{1}{3}X_{max} \\ \frac{-3}{X_{max}}X + 2 & \text{if} \quad \frac{1}{3}X_{max} < X \leq \frac{2}{3}X_{max} \\ 0 & \text{if} \quad otherwise \end{cases} \tag{4.20}$$

$$\mu_{PM}(X) = \begin{cases} \frac{3}{X_{max}}X - 1 & \text{if} \quad \frac{1}{3}X_{max} < X \leq \frac{2}{3}X_{max} \\ \frac{-3}{X_{max}}X + 3 & \text{if} \quad \frac{2}{3}X_{max} < X \leq X_{max} \\ 0 & \text{if} \quad otherwise \end{cases} \tag{4.21}$$

$$\mu_{PB}(X) = \begin{cases} 1 & \text{if} \quad X_{max} \leq X \\ \frac{3}{X_{max}}X - 2 & \text{if} \quad \frac{2}{3}X_{max} < X \leq X_{max} \\ 0 & \text{if} \quad otherwise \end{cases} \tag{4.22}$$

where X is representing one of the input variables $e(k)$, Δy_p, and Δu_p. Note that, here $X_{min} = -X_{max}$ and the input variables Δy_p, and Δu_p are considered in a single vector form of $\Delta w_p = [\Delta y_p \ \ \Delta u_p]^T$.

FGS Rules

After the fuzzy numbers representing each input variable are obtained, the fuzzy numbers of the normalized output variables $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$ are determined using the rule decision tables. The rule decision table defines the rules to update each of the normalized SPC gains. Generally, the membership degree of normalized output variable Y is obtained by the product of membership degrees of input variables X_1 and X_2 as below,

$$\mu_i(Y) = \mu_{AA_i}(X_1) \cdot \mu_{BB_i}(X_2) \tag{4.23}$$

Tables 4.1, 4.2 and 4.3 show the FGS rules to update the normalized SPC gains, $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$, respectively [135, 140]. Note that, the rules are derived experimentally based on a piecewise constant reference input.

Table 4.1: Rule decision table to update $\bar{k}_e$

		Δw_p						
		NB	**NM**	**NS**	**ZZ**	**PS**	**PM**	**PB**
$e(k)$	**NB**	B	B	B	B	B	B	B
	NM	M	B	B	B	B	B	M
	NS	S	M	M	M	M	M	S
	ZZ	S	S	S	S	S	S	S
	PS	S	M	M	M	M	M	S
	PM	M	B	B	B	B	B	B
	PB	B	B	B	B	B	B	B

Table 4.2: Rule decision table to update $\bar{k}_{\Delta y_p}$

		Δw_p						
		NB	**NM**	**NS**	**ZZ**	**PS**	**PM**	**PB**
$e(k)$	**NB**	S	S	S	S	S	S	S
	NM	M	S	S	S	S	S	M
	NS	B	M	S	S	S	M	B
	ZZ	B	B	M	M	M	B	B
	PS	B	M	S	S	S	M	B
	PM	M	S	S	S	S	S	M
	PB	S	S	S	S	S	S	S

Table 4.3: Rule decision table to update $\bar{k}_{\Delta u_p}$

		Δw_p						
		NB	**NM**	**NS**	**ZZ**	**PS**	**PM**	**PB**
$e(k)$	**NB**	B	B	B	M	M	S	S
	NM	B	B	M	M	S	S	M
	NS	B	M	M	S	S	M	M
	ZZ	M	M	S	S	S	M	M
	PS	M	M	S	S	M	M	B
	PM	M	S	S	M	M	B	B
	PB	S	S	M	M	B	B	B

Defuzzification

In defuzzifier the fuzzy number of normalized SPC gains which are obtained based on FGS rules are converted to the crisp gain values. Here, three different *triangular-shape* fuzzy subsets with linguistic fuzzy variables *big* (B), *medium* (M) and *small* (S) are defined for the normalized output variables, $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$. The membership functions of normalized output variables are shown in Figure 4.2 [135, 140].

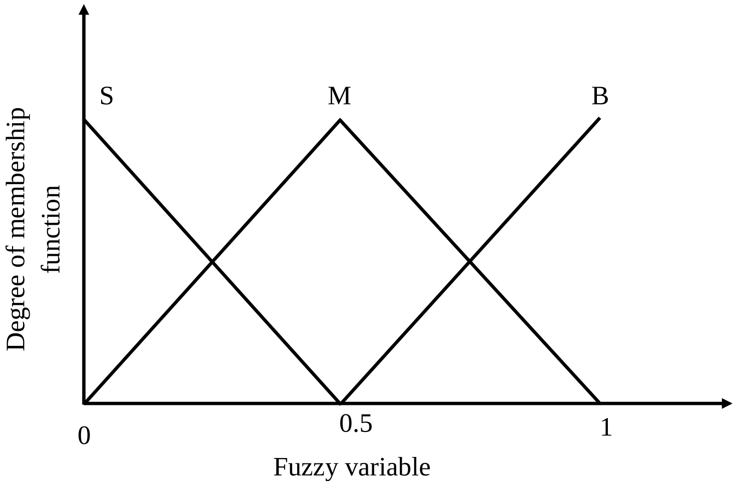

Figure 4.2: Fuzzy subsets for $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$

In general, the normalized output variable Y corresponding to $\mu_i(Y)$ is obtained from the following equation,

$$Y = \frac{\sum\limits_{i} \mu_i(Y) Y_i}{\sum\limits_{i} \mu_i(Y)} \quad , \quad i = 1, 2, \ldots, number\ of\ rules \tag{4.24}$$

where Y_i is the value of the normalized output variable Y corresponding to the $\mu_i(Y)$ for ith rule.

In the proposed FGS-SPC method, the output variables can be determined using the following defuzzification formula,

$$\bar{k}_e = \frac{\sum\limits_{i=1}^{49} \mu_i(\bar{k}_e)\bar{k}_{ei}}{\sum\limits_{i=1}^{49} \mu_i(\bar{k}_e)} \tag{4.25}$$

$$\bar{k}_{\Delta w_p(j)} = \frac{\sum_{i=1}^{49} \mu_i(\bar{k}_{\Delta w_p(j)})\bar{k}_{\Delta w_p(j)i}}{\sum_{i=1}^{49} \mu_i(\bar{k}_{\Delta w_p(j)})} \tag{4.26}$$

where $\Delta w_p(j) = [\Delta y_p(j) \;\; \Delta u_p(j)]^T$.

Once the crisp values of the normalized SPC gains, $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$, are obtained, the actual SPC gains $\tilde{k}_e$, $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$ are calculated from the following equations by having the CGRs,

$$\tilde{k}_e = (\tilde{k}_{e,max} - \tilde{k}_{e,min})\bar{k}_e + \tilde{k}_{e,min} \tag{4.27}$$

$$\tilde{k}_{\Delta y_p} = (\tilde{k}_{\Delta y_{p,max}} - \tilde{k}_{\Delta y_{p,min}})\bar{k}_{\Delta y_p} + \tilde{k}_{\Delta y_{p,min}} \tag{4.28}$$

$$\tilde{k}_{\Delta u_p} = (\tilde{k}_{\Delta u_{p,max}} - \tilde{k}_{\Delta u_{p,min}})\bar{k}_{\Delta u_p} + \tilde{k}_{\Delta u_{p,min}} \tag{4.29}$$

Figure 4.3 shows block diagram of the PSO-based FGS-SPC system.

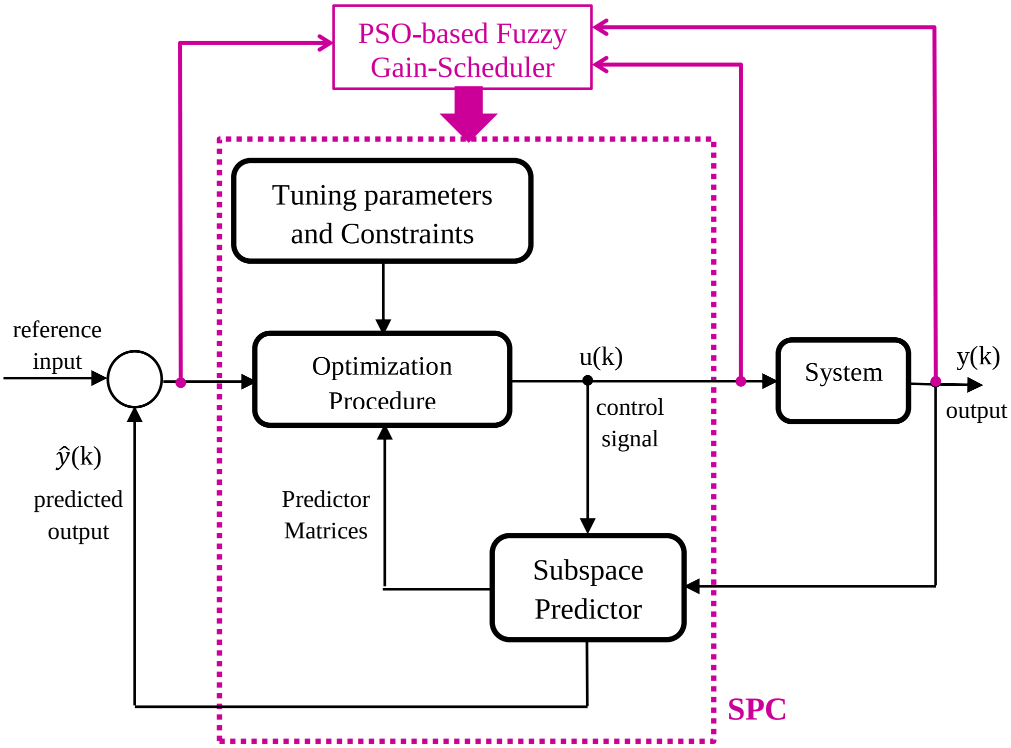

Figure 4.3: SPC system with PSO-based FGS

4.3.2 PSO-based FGS Technique

In FGS procedure, the CGRs in (4.10), (4.11) and (4.12) completely depend on the process and may vary by variation of the process parameters [112]. In conventional FGS, the CGRs are usually determined by rule of thumb and the designer's experience at the beginning of the FGS procedure [117], which is a time consuming procedure, especially for time-varying and nonlinear systems. Here, we proposed a PSO-based FGS-SPC technique to optimally-automatically calculate the CGRs and update the SPC gains $\tilde{k}_e$, $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$ [140].

By utilizing the aforementioned *Constriction PSO* algorithm in (2.177) and (2.178), the CGRs in (4.10), (4.11) and (4.12) can be selected as PSO particles. All PSO particles and parameters are initialized according to PSO commendations. At each iteration, the controller gains $\tilde{k}_e$, $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$ are calculated based on particle information and defuzzification results from the FGS algorithm. Next, the calculated controller gains are used to evaluate the PSO fitness function. Here, SPC cost function in (2.117) is selected as the PSO fitness function. This selection enables us to optimize the CGRs in FGS and SPC cost function, simultaneously. Finally, the global best is updated based on the evaluation results. The algorithm continues until the convergence condition is met. Here, an iteration limit is considered to terminate the PSO algorithm. Moreover, to avoid the increase of the computational time, a criterion is set to terminate the algorithm when there is no significant changes in particle movements. Algorithm 4.3.1 presents the PSO-based FGS-SPC procedure [140].

Remark 4.3.1. *Unlike the existing FGS algorithms in PSO-based FGS technique, the CGRs in (4.10), (4.11) and (4.12) are not manually selected as constant values, but they are calculated and updated optimally-automatically at each iteration according to the PSO fitness function. Therefore, the proposed PSO-based FGS technique by itself is a valuable package, which can be applied to any control algorithm that requires CGRs tuning for FGS.*

Remark 4.3.2. *Note that since the re-arranging of SPC control law in Section 4.2 has been done by considering a piecewise constant reference input; therefore, to met the PSO convergence condition the input constant intervals need to be not smaller than the PSO convergence time.*

Algorithm 4.3.1. *PSO-based FGS-SPC Method*

Phase I: Calculation of normalized SPC gains via FGS procedure

Inputs: Tracking error, $e(k)$, variation of past output data Δy_p and variation of past control signal Δu_p

Outputs: Normalized SPC gains, $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$

Step 1: Fuzzification using input membership functions in Figure 4.1

Step 2: Applying fuzzy rules from Tables 4.1, 4.2, 4.3

Step 3: Defuzzification using output membership functions in Figure 4.2 to obtain the normalized SPC gains $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$

Phase II: Calculate the optimum CGRs via PSO Procedure

Inputs: Normalized SPC gains, $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$ from Phase I, and the presumed CGRs, $\tilde{k}_{e,min}$, $\tilde{k}_{e,max}$, $\tilde{k}_{\Delta y_p,min}$, $\tilde{k}_{\Delta y_p,max}$, $\tilde{k}_{\Delta u_p,min}$ and $\tilde{k}_{\Delta u_p,max}$

Outputs: The optimum CGRs

Step 1: Initialize PSO parameters and set CGRs as particles

Step 2: Calculate SPC gains based on CGRs info and the normalized SPC gains from Phase I using (4.27), (4.28), (4.29)

Step 3: Evaluate SPC cost function and update the personal best and the global best

Step 4: Check the convergence (the iteration limit or particle movement threshold). If YES goto Phase III. If NO update the CGRs using the new global best and personal best then goto Step 2

Phase III: Calculate actual SPC gains

Inputs: Normalized SPC gains from Phase I, $\bar{k}_e$, $\bar{k}_{\Delta y_p}$ and $\bar{k}_{\Delta u_p}$ and the optimum CGRs from Phase II

Outputs: The SPC gains $\tilde{k}_e$, $\tilde{k}_{\Delta y_p}$, $\tilde{k}_{\Delta u_p}$

Step 1: Calculate SPC gains based on optimally selected CGRs from Phase II by using (4.27), (4.28), (4.29)

4.4 Simulation Results

In this section, we demonstrate the performance and efficiency of proposed PSO-based FGS-SPC technique by applying it to the SISO Dryer System [4, 141].

4.4.1 System Description

Dryer System: SISO Minimum phase fourth-order system

The proposed PSO-based FGS-SPC method was tested on data set collected from a SISO laboratory scale “hairdryer” system [4, 141]. In this system, air is fanned through a tube and heated at the inlet. Input of the system is the voltage over the heating device, which is a mesh of resistor wires. Output of the system is the air temperature, which is measured by a thermocouple (or rather the voltage from the thermocouple). The sampling time was set to $T_s = 0.08sec$ throughout the simulation. Figure 4.4 shows the collected open-loop I/O data for this system from [20].

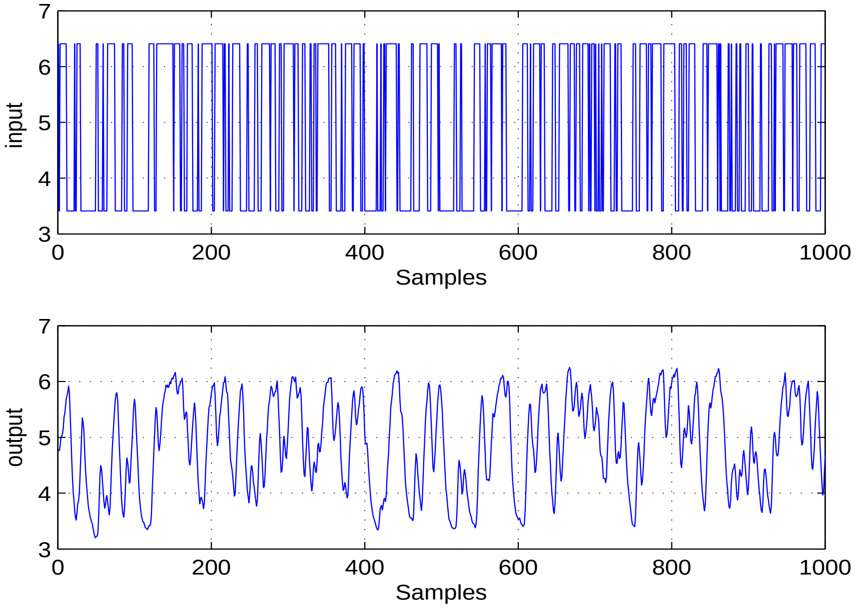

Figure 4.4: Dryer System: Collected input-output data from open-loop system

The open-loop system can be identified as a forth order system by applying subspace identification method [20]. Eigenvalues of the system are located at,

$$\lambda_{1,2} = 0.4735 \pm j0.2037, \quad \lambda_3 = 0.7439, \quad \lambda_4 = 0.9475$$

Remark 4.4.1. *In PSO-based FGS-SPC algorithm there is no need to identify the open-loop system. Here, the eigenvalues are only provided to test the robustness of the proposed method in simulation.*

4.4.2 Stability Analysis

First, the technique in Table 3.1 is applied to determine the FR of N_p and N_{SFP}. The DC-gain is determined from the open-loop I/O data as $K_{ol} = 0.97802 > 0$. Figure 4.5 shows the SPC stability graph for this system.

According to the SPC stability graph and the DC-gain, we have $N_{SFP} = 4$ and FR of N_p is $4 \leq N_p \leq M$. In this simulation, the number of row blocks in data Hankel matrices has been chosen as $M = 30$. Then, the efficient values of prediction horizon and control horizon are determined by using the technique in Table 3.2 as $N_{EC} = 5$ and $N_{EP} = 14$.

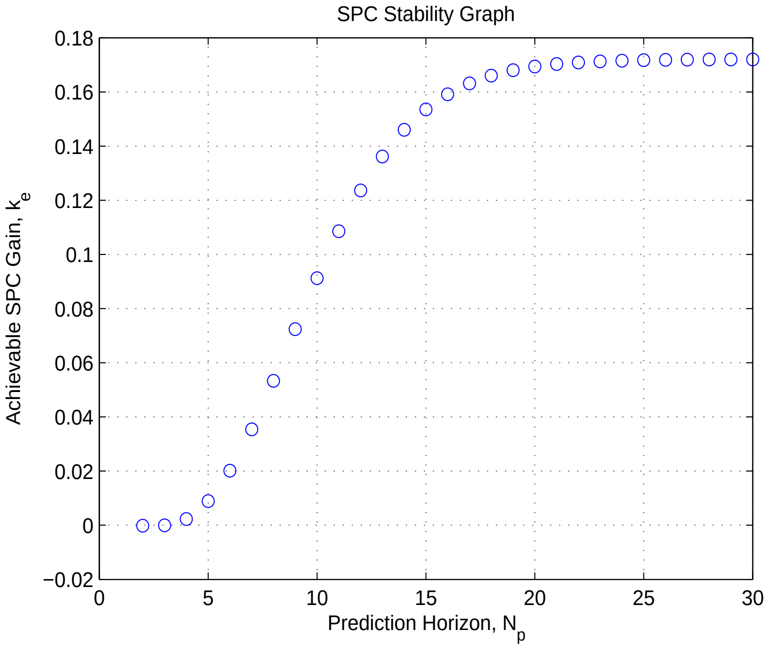

Figure 4.5: Dryer System: SPC stability graph for $Q = I$, $R = 30I$ and $N_c = N_p - 1$

4.4.3 Simulation Initialization

The appropriate choice of the initialization parameters has a significant impact on the stability and performance of the designed control system. In this simulation, the controller parameters are considered as follows:

- SPC parameters and constraints:

$$N_p = N_{EP} = 14\ ,\quad N_c = N_{EC} = 5\ ,\quad Q = I\ ,\quad R = 30I\ ,\quad -10 \leq u(k) \leq 10\ ,\quad -1 \leq \Delta u(k) \leq 1$$

- FGS parameters:

$$e_{max} = -e_{min} = 50\quad ,\quad \Delta y_{pmax} = -\Delta y_{pmin} = 1\quad ,\quad \Delta u_{pmax} = -\Delta u_{pmin} = 0.5$$

- PSO parameters:

$$\kappa = 1\quad ,\quad \phi_1 = \phi_2 = 2.05\quad ,\quad population\ size = 100\quad ,\quad iteration\ limit = 100$$

In order to eliminate the adverse effect of fluctuations at the output signal, specifically for SPC method, a user-defined tolerance, $\epsilon = 0.1$, is applied to the tracking error.

4.4.4 Robustness and Adaptiveness Test

To study the adaptability of the controllers in the presence of the time-varying systems, we assume that there are some changes in the eigenvalues of the system at sampling instants $k = 200$ and $k = 800$, respectively, as follows:

$$\lambda_{1,2} = 0.9238 \pm j0.2441, \quad \lambda_3 = 0.2939, \quad \lambda_4 = 0.9468$$

$$\lambda_{1,2} = 0.9486 \pm j0.2042, \quad \lambda_3 = -0.6184, \quad \lambda_4 = 0.6096$$

Figure 4.6 shows the input and output of the closed-loop system for PSO-based FGS-SPC, FGS-SPC and SPC methods with previously mentioned constraints on control signal, $u(k)$, and its variation $\Delta u(k)$.

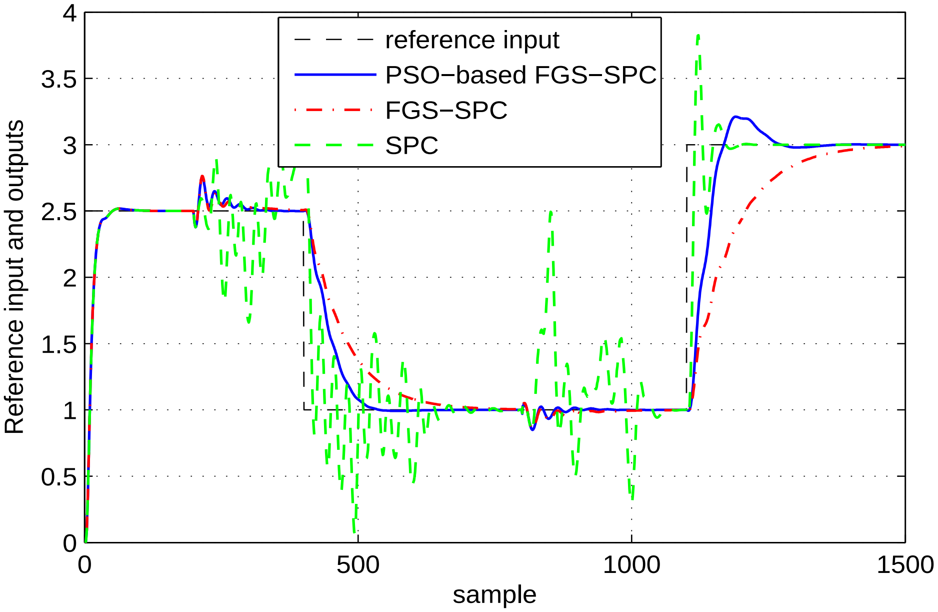

Figure 4.6: Dryer System: Reference input and outputs for PSO-based FGS-SPC, FGS-SPC and SPC

According to the results, by changing the system dynamics at $k = 200$ and $k = 800$, all three aforementioned methods start to adapt their controllers with the new conditions. However, Figure 4.6 shows that SPC starts applying PE input signals to identify the new model of the system to update the subspace predictor matrices L_u and L_w, and then calculates the new controller gains $\tilde{k}_e$ and $\tilde{k}_{\Delta w_p}$. It can be seen that this process makes large fluctuations in the output of the system and takes a lot of time to capture the new model. Also, overshoots and undesired fluctuations at the step response indicates that SPC also needs to update and re-tune the weighting matrices Q and R to obtain a smooth response for new systems.

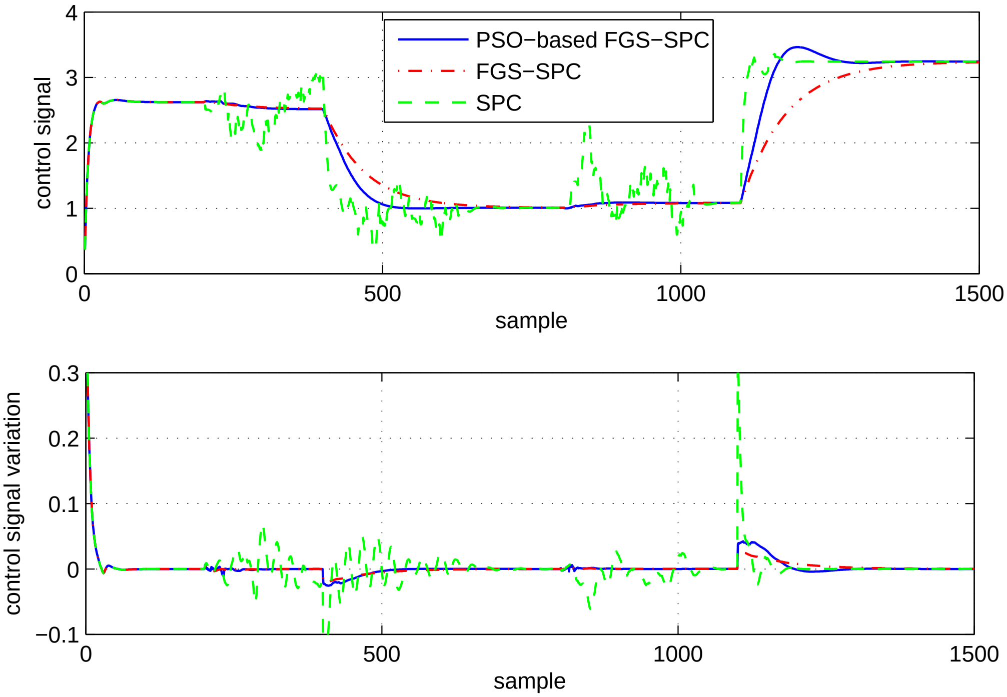

Figure 4.7: System I: Control signal u, and its variation Δu for PSO-based FGS-SPC, FGS-SPC and SPC

Control signal, $u(k)$, and its variation, $\Delta u(k)$, are shown in Figure 4.7 for each of the control algorithms. The results in Figures 4.6 and 4.7 show that both PSO-based FGS-SPS and FGS-SPC methods can adapt the controller gains, $\tilde{k}_e$, $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$, much faster than the SPC with less fluctuation, and manage to comply with the specified constraints on $u(k)$ and $\Delta u(k)$. Moreover, PSO-based FGS-SPC has faster response than the FGS-SPC method due to optimally tunning of the CGRs by using the PSO algorithm.

Figures 4.8, 4.9 and 4.10 demonstrates how the controller gains $\tilde{k}_e$, $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$ were adjusted for each of the above-mentioned methods.

The fluctuations in tuning of the controller gains for the SPC method is observable in Figure 4.10, which is because of applying PE signals in SPC. Figure 4.9 shows that FGS-SPC technique arranged the controller gains inside a small boundary to ensure the stability of the algorithm

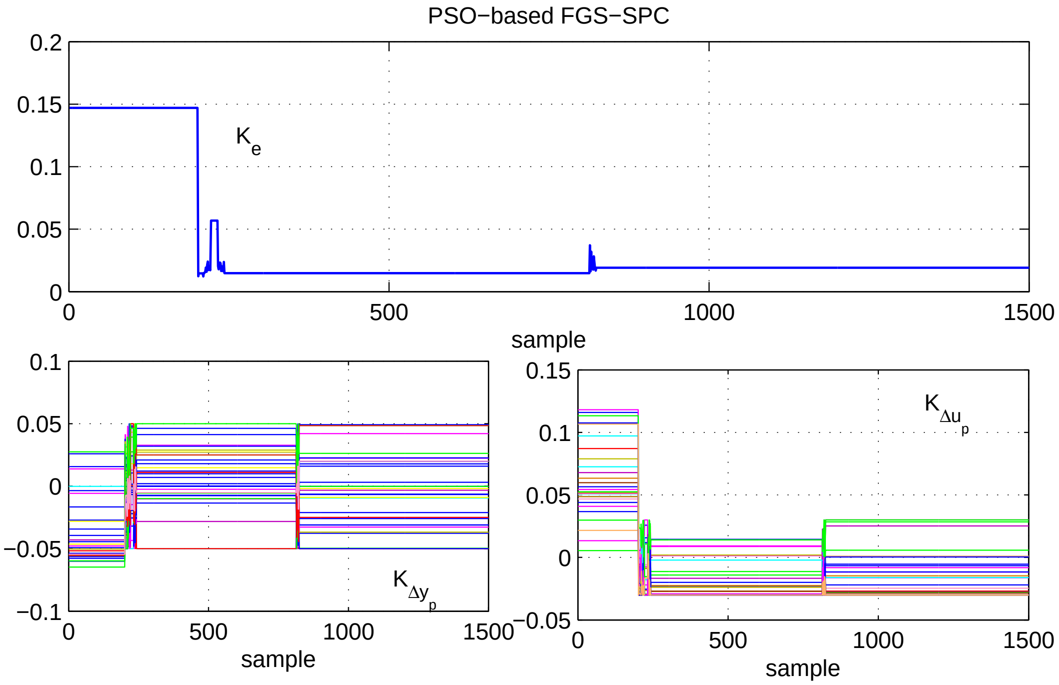

Figure 4.8: Dryer System: Updating the controller gains for PSO-based FGS-SPC method

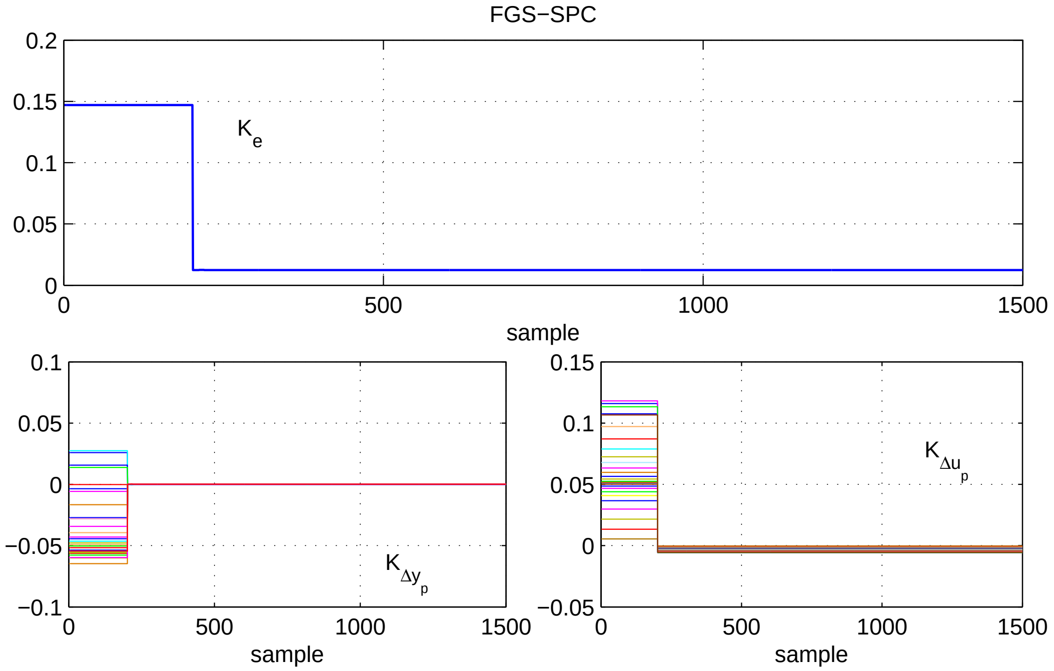

Figure 4.9: Dryer System: Updating the controller gains for FGS-SPC method

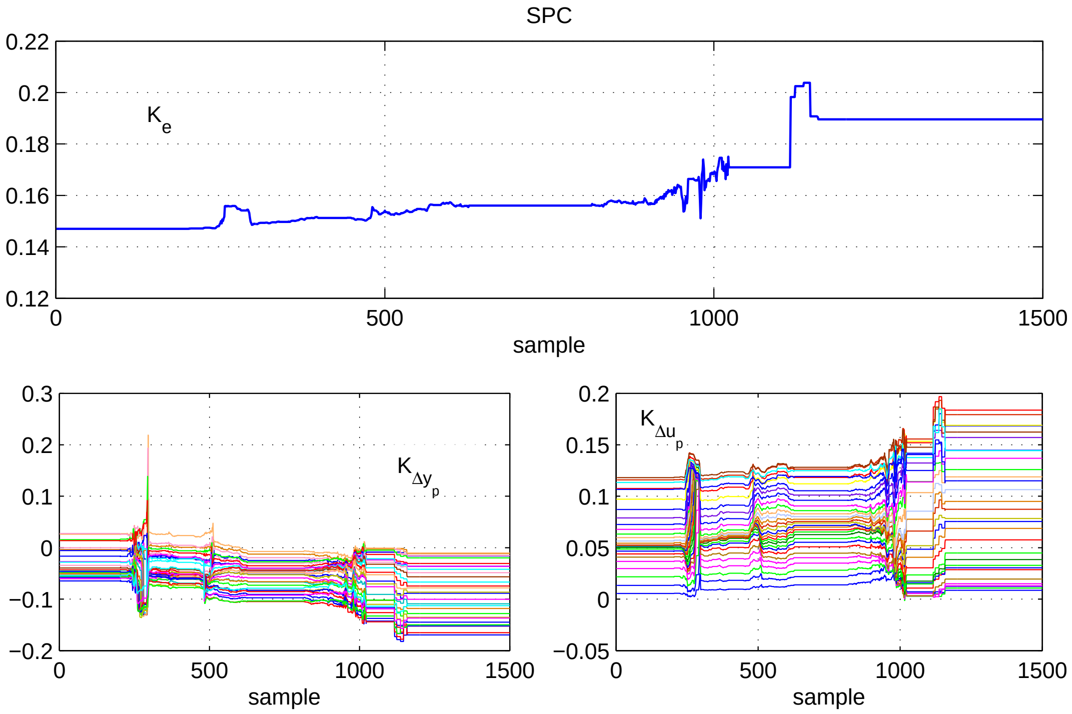

Figure 4.10: Dryer System: Updating the controller gains for SPC method

without considering its optimality. Figure 4.8 shows flexibility of the proposed PSO-based FGS-SPC method in optimally tuning the controller gains compared to the FGS-SPC method in Figure 4.9. Consequently, PSO-based FGS-SPC can provide both optimal and stable closed-loop system by utilizing the PSO technique.

To investigate the robustness of the proposed PSO-based FGS-SPC algorithm in the presence of noisy data, a Gaussian noise with $SNR = 25db$ is added to the I/O data in simulations. Also, the user-defined tolerance is changed to $\epsilon = 0.4$ to eliminate the adverse effect of fluctuations in the presence of noisy data. Figure 4.11 illustrates the input and output of the closed-loop noisy system for each aforementioned method. Although all three methods can adapt the controller gains according to their control algorithm, but having a large overshoot in the response of the SPC method shows that it needs to re-tune the weighting matrices in the SPC cost function. PSO-based FGS-SPC has faster response compared to FGS-SPC, and less fluctuations compared to SPC method that shows its superiority with respect to other methods.

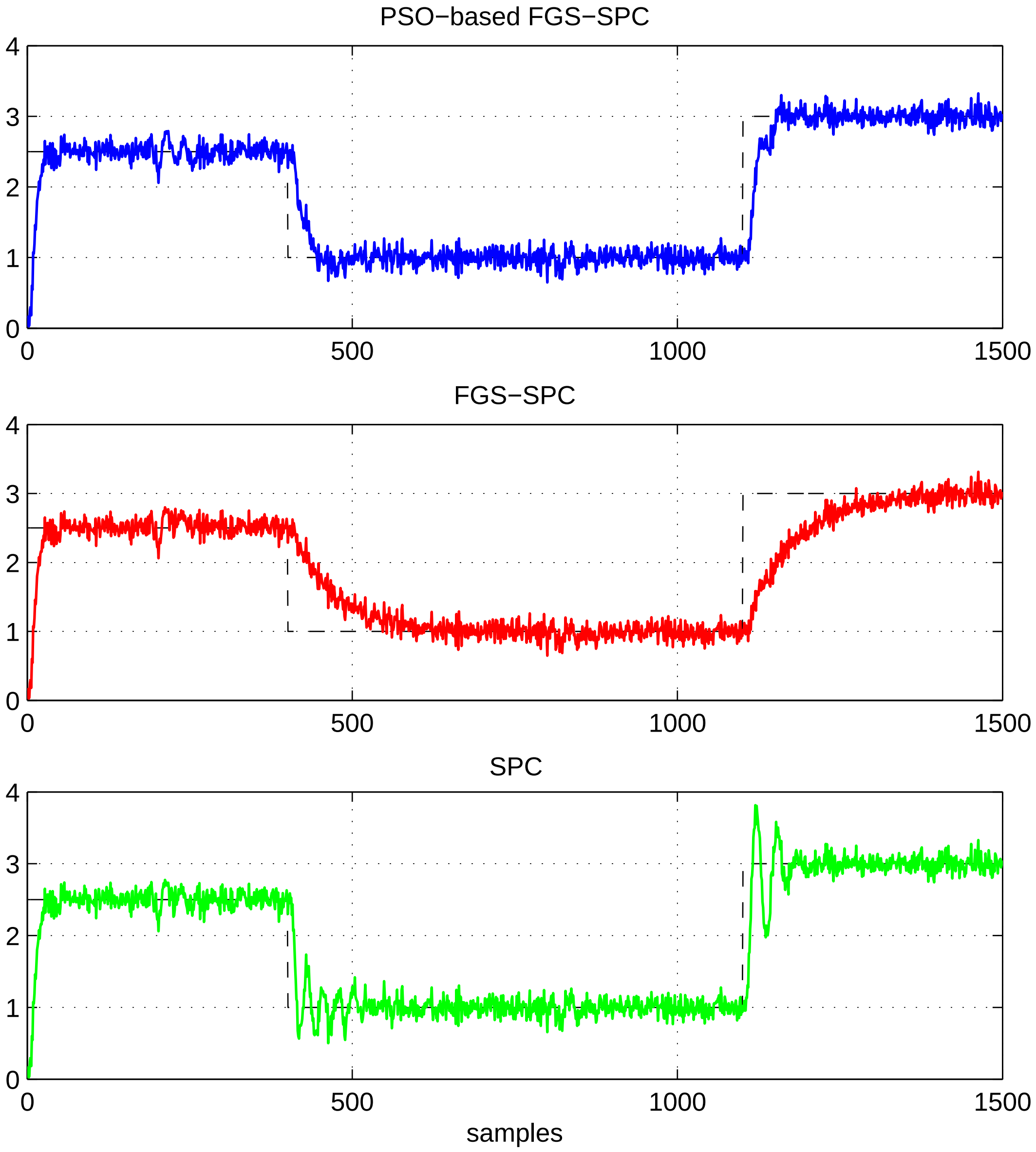

Figure 4.11: Dryer System: Reference input (dashed) and outputs for PSO-based FGS-SPC, FGS-SPC and SPC methods for noisy system

4.4.5 Performance Evaluation

In order to compare the performance of each method, performance points are depicted in Figure 4.12. The results show that PSO-based FGS-SPC has the best performance among these three methods. PSO-based FGS-SPC has smaller tracking error variance and variation of control signal variance than SPC method which shows superiority of the proposed method. In comparison with FGS-SPC, the PSO-based FGS-SPC can significantly decrease the variance of the tracking error, but it has a small amount of increase in the variance of the incremental control signal.

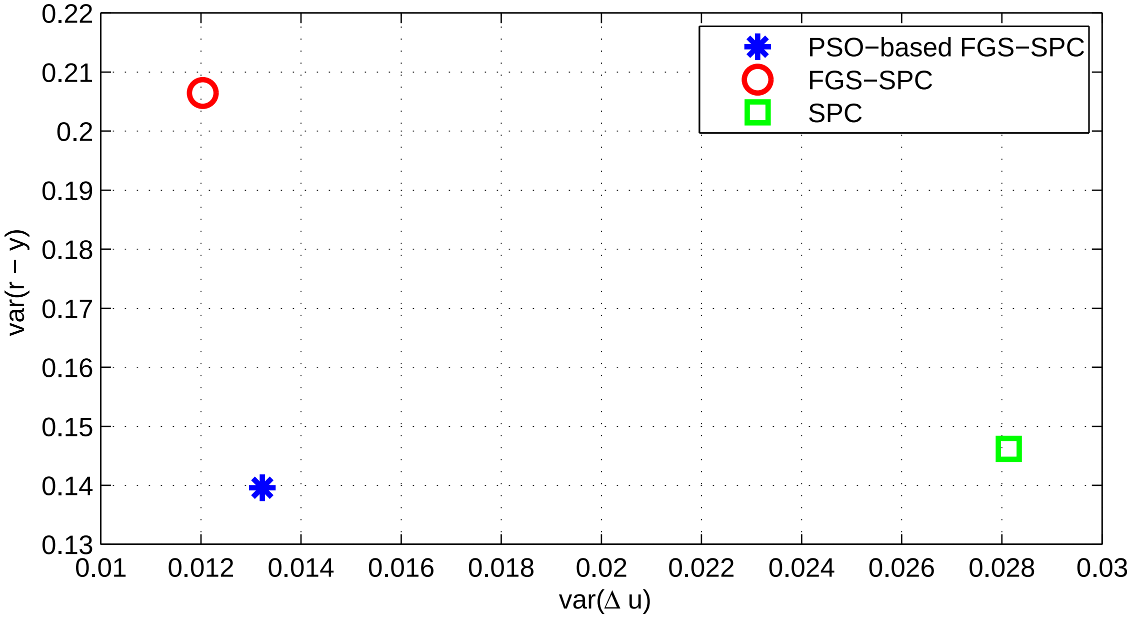

Figure 4.12: Dryer System: Performance points for each method

To show the optimality of proposed PSO-based FGS-SPC method, cost function evaluation, J of the methods for each sample time is depicted in Figure 4.13 as the performance index. Convergence results of the performance index in Part(a) and Part(b) of the Figure 4.13 are shown in Table 4.4. The results show effectiveness and superiority of the proposed PSO-based FGS method with respect to SPC and FGS-SPC methods, and superiority of the proposed technique in terms of optimality.

Although SPC method shows faster response than the other two methods, it has large deviation at the beginning and slow convergence pace due to the fluctuation. These fluctuations are generated by applying PE signals, and SPC needs to re-tune its parameters, such as weighting matrices and horizons in the cost function. In some cases, one of which has been shown in Figure 4.14, the fluctuations can result a non-convergence situation for the SPC method.

Table 4.4: Dryer System: Performance index value for each method

Performance Index	PSO-based FGS-SPC	FGS-SPC	SPC
Figure 4.13(a)	5.87×10^{-16}	4.33×10^{-12}	5.21×10^{-7}
Figure 4.13(b)	1.82×10^{-16}	1.06×10^{-4}	1.35×10^{-16}

To demonstrate the computational time saving feature of the proposed PSO-based FGS-SPC

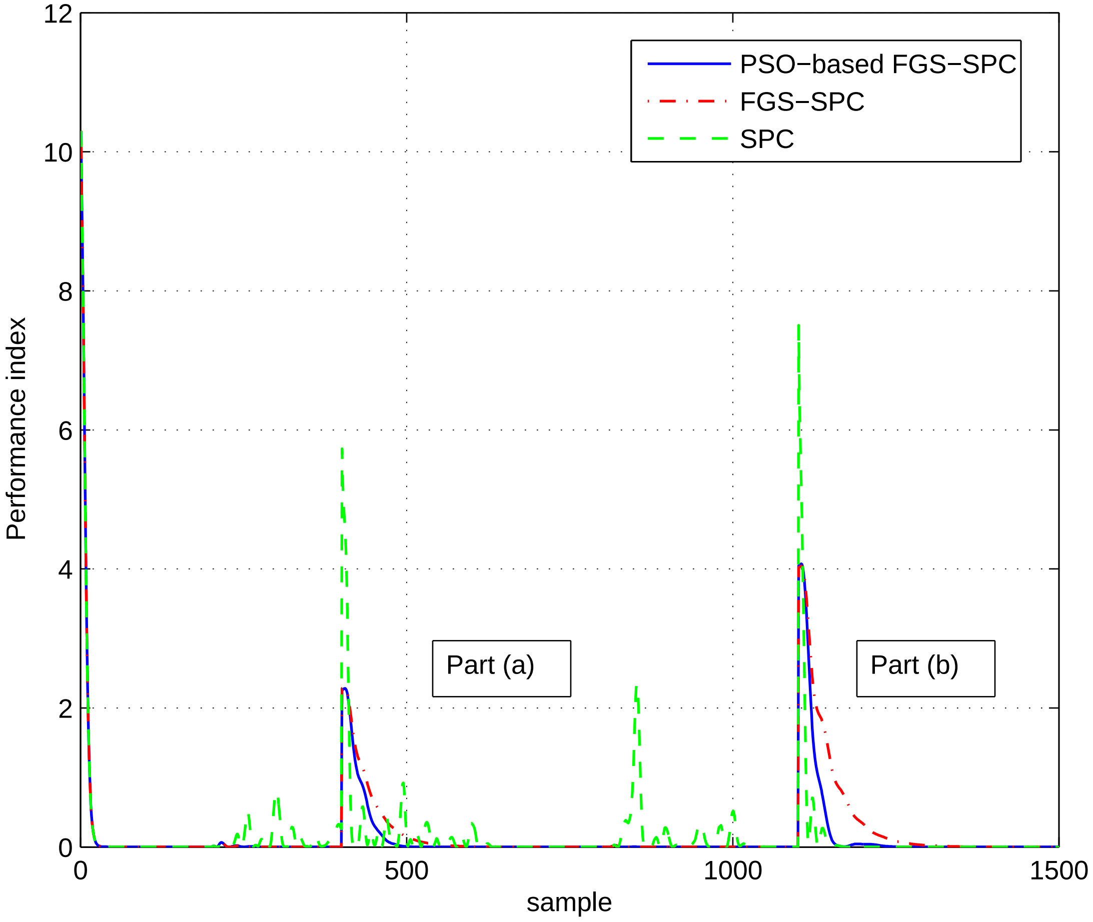

Figure 4.13: Dryer System: Performance index value for each of the methods

algorithm, Monte-Carlo simulations consisting of 30 runs were conducted for all methods with different row dimensions of data Hankel matrices, M. Figure 4.15 shows the average computational time that is required to complete the updating algorithm through the MATLAB programming.

The results show that the computational load increases by increasing the row dimension of the past and future data Hankel matrices. However, the PSO-based FGS-SPC algorithm needs much less computational time than SPC method. In addition, it shows that time efficiency of the PSO-based FGS-SPC is much less sensitive to the increasing of Hankel matrix orders. Moreover, Figure 4.15 shows PSO-based FGS-SPC needs more computational time than the FGS-SPC method, which is because of applying the PSO procedure. However, Figure 4.15 also shows that by increasing the M, the gap between two graphs decreases. As a result, for large values of M, the PSO algorithm is computationally more efficient.

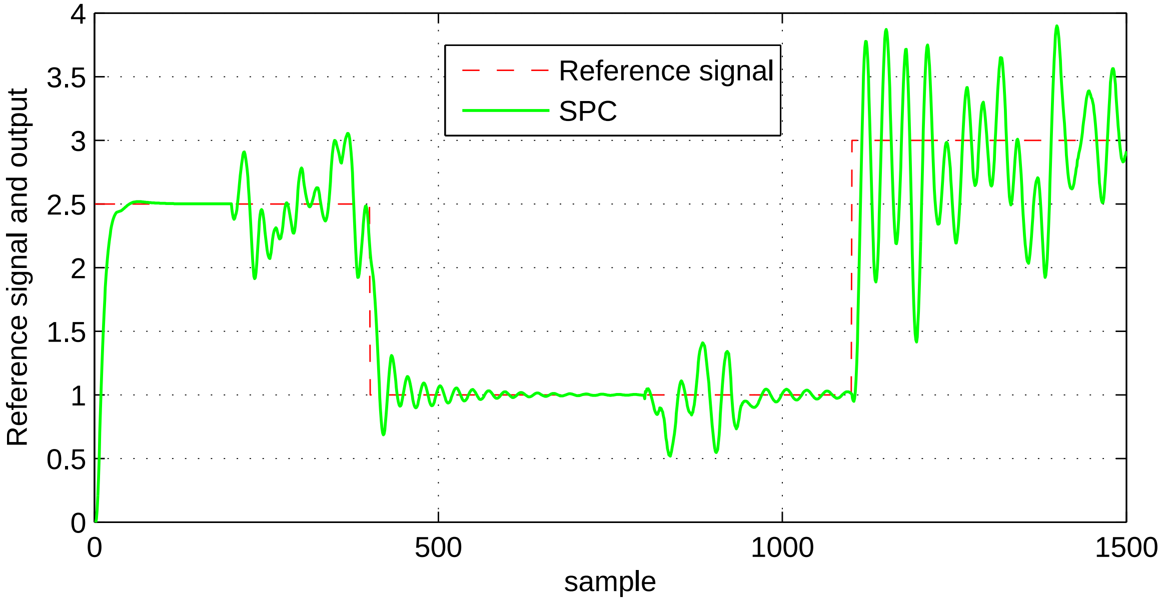

Figure 4.14: Dryer System: Reference input and outputs for SPC method in non-convergence case

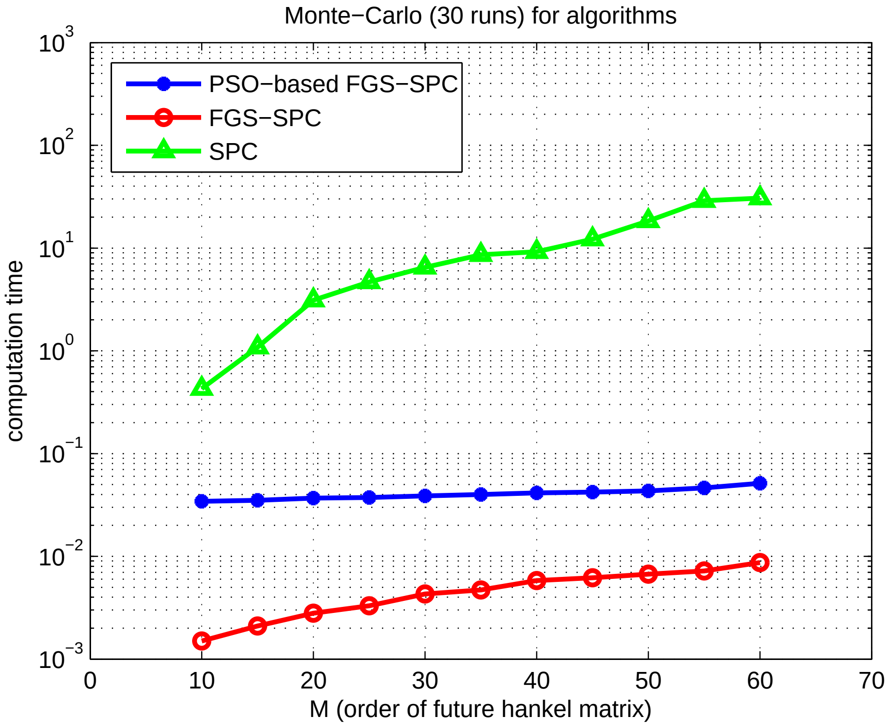

Figure 4.15: Dryer System: Monte-Carlo simulations consisting of 30 runs for PSO-based FGS-SPC and SPC

4.5 Summary

In this chapter, an algorithm has been provided to eliminate the requirement of disruptive and time consuming process of applying PE signals in SPC. The proposed method, denoted by PSO-based FGS-SPC, is based on optimally-automatically tuning of SPC controller gains by applying the proposed PSO-based FGS technique. In this technique, the CGRs of the FGS algorithm are calculated and updated optimally by applying the PSO technique. In PSO-based FGS-SPC, by considering the SPC cost function as PSO fitness function, the SPC controller gains and the CGRs of FGS algorithm are simultaneously tuned. Moreover, the proposed PSO-based FGS technique by itself is a valuable package, which can be applied to any control algorithm that requires CGRs tuning for FGS. Robustness and efficiency of the method have been illustrated by simulation on a time-varying SISO system in the presence of constraints and noisy data. Simulation results indicate the superiority of the proposed PSO-based FGS-SPC method over FGS-SPC and SPC in achieving an optimal solution and in providing a fast and robust tracking performance. In addition, the main advantage of PSO-based FGS-SPC over SPC is that it does not require applying disruptive PE signals; therefore, it provides a time-efficient control strategy.

Chapter 5

Time-delay and Order Selection for SIM and SPC

5.1 Problem Statement and Chapter Summary

Since SPC is a *black-box* data-driven approach, it has no need to collect prior knowledge about the system for designing the controller, making it appropriate for automatic control purposes. However, prediction results and the closed-loop performance can be deteriorated in the presence of noisy I/O data. In this scenario, integrating a priori information about the system characteristics, such as time-delay and system order can help to appropriately select the SPC parameters and enhance the performance. Moreover, order selection and time-delay estimation are still open problems in the SIM framework. There are some threshold-based techniques in the SIM framework to estimate these parameters, which are discussed in Section 2.3. However, the estimation of these parameters from noisy I/O data by threshold-based techniques is a tricky task, and sometimes utilizing more powerful statistic methods would be helpful.

In this chapter, we show that using the new approach, denoted by RE-based TDE [142], is a robust method of time-delay estimation that outperforms the existing approaches of time-delay estimation. In addition, we propose to utilize a new method denoted by MSEE [143] for order estimation and show the advantages of this method over the conventional thresholding approaches. It is shown that MSEE also outperforms existing non-thresholding methods.

5.2 Data-driven Approaches for Time-delay Estimation and Order Selection

In this section Reconstruction Error based (RE-based) method and Mean Squared Eigenvalue Error Method (MSEE) are presented for time-delay estimation and order selection for SPC design in the presence of noisy I/O data.

5.2.1 Reconstruction Error based (RE-based) Method

The *Reconstruction Error* (RE) has been introduced in [144] as below,

$$e_{RE} \;=\; \frac{1}{N}||\bar{y}-\hat{y}||_2^2 \tag{5.1}$$

which is the error between the N samples of *true unavailable* noise-free output data, $\bar{y}$, and the estimated output, $\hat{y}$.

The RE-based time-delay estimation (RE-based TDE) method is an statistical data-driven method proposed to estimate the RE for a possible range of time-delays ($0 \leq N_d \leq N$) [142]. Then it determines the optimum time-delay as the one that minimizes the upper bound of the RE in (5.1). The RE has been used in finding the optimum length of the estimated impulse response coefficients (IRCs) in [144]. In the RE-based TDE, the same approach is utilized to estimate the optimum time-delay simultaneously.

Therefore, in the RE-based TDE, the optimum IRCs length and the time-delay can be estimated by solving the following minimization problem [142],

$$(N_d^*, L_h^*) = \arg\min_{N_d, L_h} \overline{e_{RE}} \tag{5.2}$$

where N_d^* is the optimum time-delay estimation, L_h^* is the optimum length of IRCs and $\overline{e_{RE}}$ is the upper bound of the RE in (5.1), which is defined in [142, 144].

Application of RE-based TDE Method for Time-delay Estimation in SIM

In Section 2.3, the Subspace-based approach for time-delay estimation has been shown by using the estimated impulse/step response coefficients (IRCs and SRCs) in (2.166) and (2.167) by selecting an appropriate threshold. However, in noisy scenarios the estimated IRCs and SRCs are corrupted by noise; therefore, estimating the time-delay is a tricky task. This issue can be resolved by applying the presented RE-based TDE method to the noisy IRCs of the system. Therefore, correct estimation of the system time-delay by the RE-based TDE method can improve the performance of the designed SPC for time-delay systems.

5.2.2 Mean Squared Eigenvalue Error (MSEE) Method

MSEE method is a statistical data-driven approach to denoise the eigenvalues, which are derived from noisy data [143, 145]. The optimum order selection is based on the calculation of eigenvalue estimates using the available noisy data, then sorting these estimated eigenvalues and thresholding them with a consistent criterion. The proposed criterion is obtained by minimizing the following reconstruction error (RE) [143],

$$e_{RE} \;\; = \;\; \frac{1}{N}||\bar{\lambda} - \hat{\lambda}_m||^2 \tag{5.3}$$

where $\bar{\lambda}$ is the vector of the *true unavailable* sorted noise-free eigenvalues and $\hat{\lambda}_m$ is a vector with the first m largest estimated eigenvalues and zero for the rest of the vector. The MSEE method determines the optimum order of the system as the one that minimizes the upper bound of the RE in (5.3). The RE has been first introduced in [144] to determine the optimum length of the estimated IRCs.

Therefore, in the MSEE method the optimum order of system can be estimated by solving the following minimization problem [143, 145],

$$m^* = \arg\min_m \overline{e_{RE}} \tag{5.4}$$

where m^* is the optimum order selection and $\overline{e_{RE}}$ is the upper bound of the RE in (5.3).

Application of MSEE for Order Estimation in SIM

According to Section 2.3, in SIM framework the system order is defined as the number of dominant singular values of the subspace predictor matrix L_w in (2.168). However, in the presence of noisy data, L_w is not a rank deficient matrix. Since noise magnifies the small singular values of L_w, estimating the system order by selecting an appropriate threshold is not straightforward. In these scenarios, motivated by the MSEE method, the system order can be estimated with sufficiently accuracy for SPC design by applying the presented MSEE method to the noisy singular values of matrix L_w.

5.3 Simulation Results

In this section, the effectiveness of applying the proposed RE-based TDE and MSEE methods in the SIM framework are presented on time-delayed SISO and MIMO processes with noisy data. Moreover, the superiority of presented RE-based TDE and MSEE methods are illustrated over other existing techniques by utilizing them to estimate the time-delay and system order in noisy scenarios.

5.3.1 Systems Description

System IV: SISO time-delayed noisy system

Consider the following transfer function of System IV in Section 3.4.1 with additive noise [55],

$$Y(s) = \frac{e^{-50s}}{(150s+1)(25s+1)}U(s) + E_w(s) \tag{5.5}$$

where E_w is a Gaussian noise with zero mean and variance of σ^2. Appropriate variances are selected for simulation to obtain the desired SNRs.

Distillation System: MIMO time-delayed system with disturbance

Following disturbance model is added to the Distillation System model in Section 3.4.1 [53],

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{16.7s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} + \begin{bmatrix} \frac{3.8e^{-8s}}{10.9s+1} \\ \frac{4.9e^{-3s}}{13.2s+1} \end{bmatrix} E_d(s) \tag{5.6}$$

where E_d is an independent white noise with covariance of $diag\{4.4582,\ 4.9898\}$. The discretized model of the disturbance channel (by using zero order holder) with $T_s = 1min$ is obtained as,

$$G_d(z) = \begin{bmatrix} z^{-9}\frac{0.331}{1-0.9123z^{-1}} \\ z^{-4}\frac{0.3575}{1-0.927z^{-1}} \end{bmatrix} \tag{5.7}$$

5.3.2 Impulse/Step Response Estimation from Noisy Data

Estimated IRCs and SRCs for both systems are obtained by using the Subspace-based approach in (2.166) and (2.167).

System IV: SISO time-delayed noisy system

Figures 5.1 and 5.2 show the estimated IRCs and SRCs for System IV from noisy data for $SNR =$ 0db and $SNR =$ 10db, respectively.

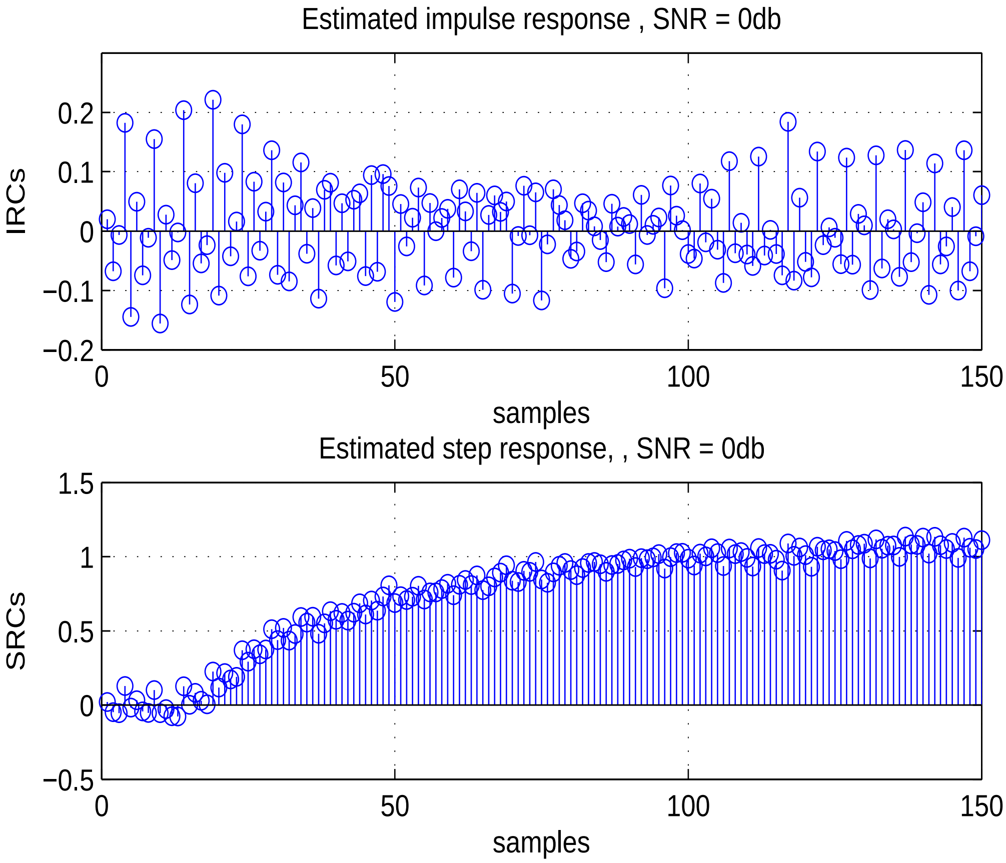

Figure 5.1: System IV: Estimated IRCs and SRCs from noisy I/O data for SNR=0db

Distillation System: MIMO time-delayed system with disturbance

Figures 5.3 and 5.4 show the estimated IRCs and SRCs for each subsystem of the Distillation System with disturbance by using the Subspace-based approach in (2.166) and (2.167). Here, each subsystem of the MIMO system, G_{pij} is denoted according to the description in Remark 3.2.1.

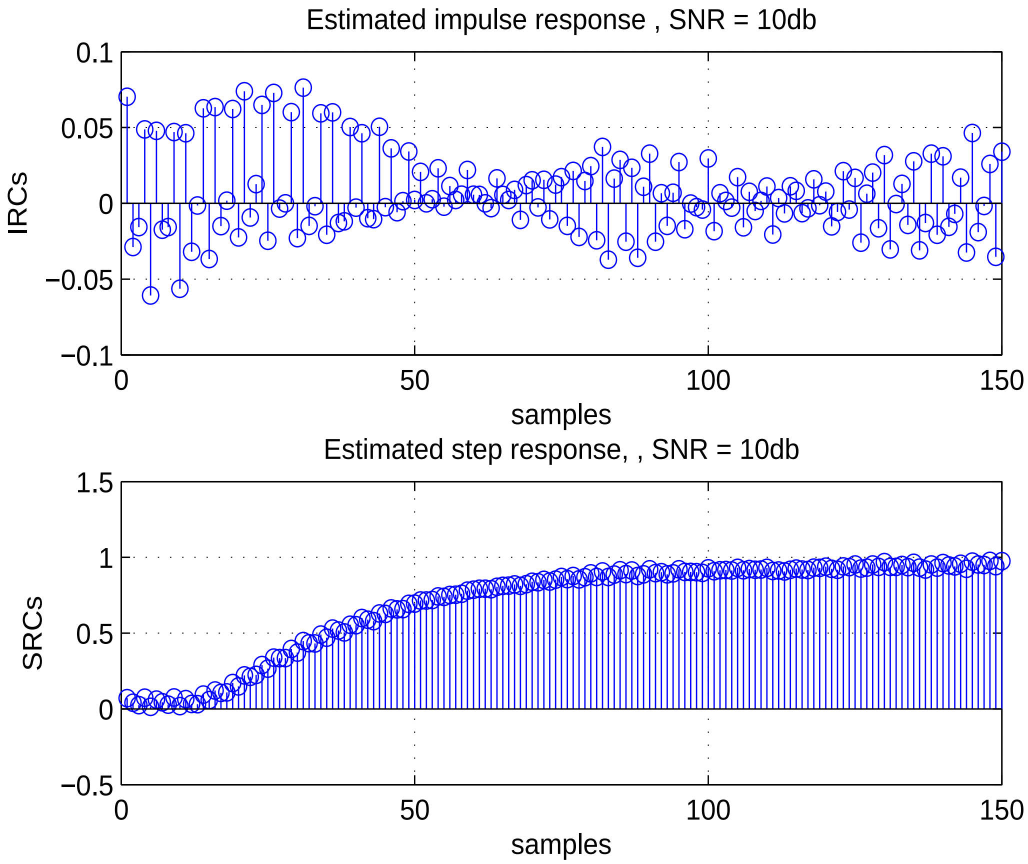

Figure 5.2: System IV: Estimated IRCs and SRCs from noisy I/O data for SNR=10db

5.3.3 Time-delay Estimation

In the presence of noisy data, the first N_d terms of estimated IRCs and SRCs are corrupted by noise. Therefore, selecting an appropriate *threshold* to estimate the time-delay is not straightforward, and threshold-based methods may not be applicable. Here, the RE-base TDE method presented in Section 5.2.1 is applied to estimate the IRCs and the time-delay.

System IV: SISO time-delayed noisy system

Figures 5.5 and 5.6 show the RE-based estimated IRCs for System IV. True IRCs and Subspace-based estimated IRCs are also plotted to compare.

It can be seen that the presented RE-based method provides better estimation of IRCs than Subspace-based approach in Section 2.3, and it can estimate the correct time-delay of $N_d^* = 11$ for both SNR cases.

Moreover, Table 5.1 shows the estimated time-delay for noisy System IV in (5.5) via different

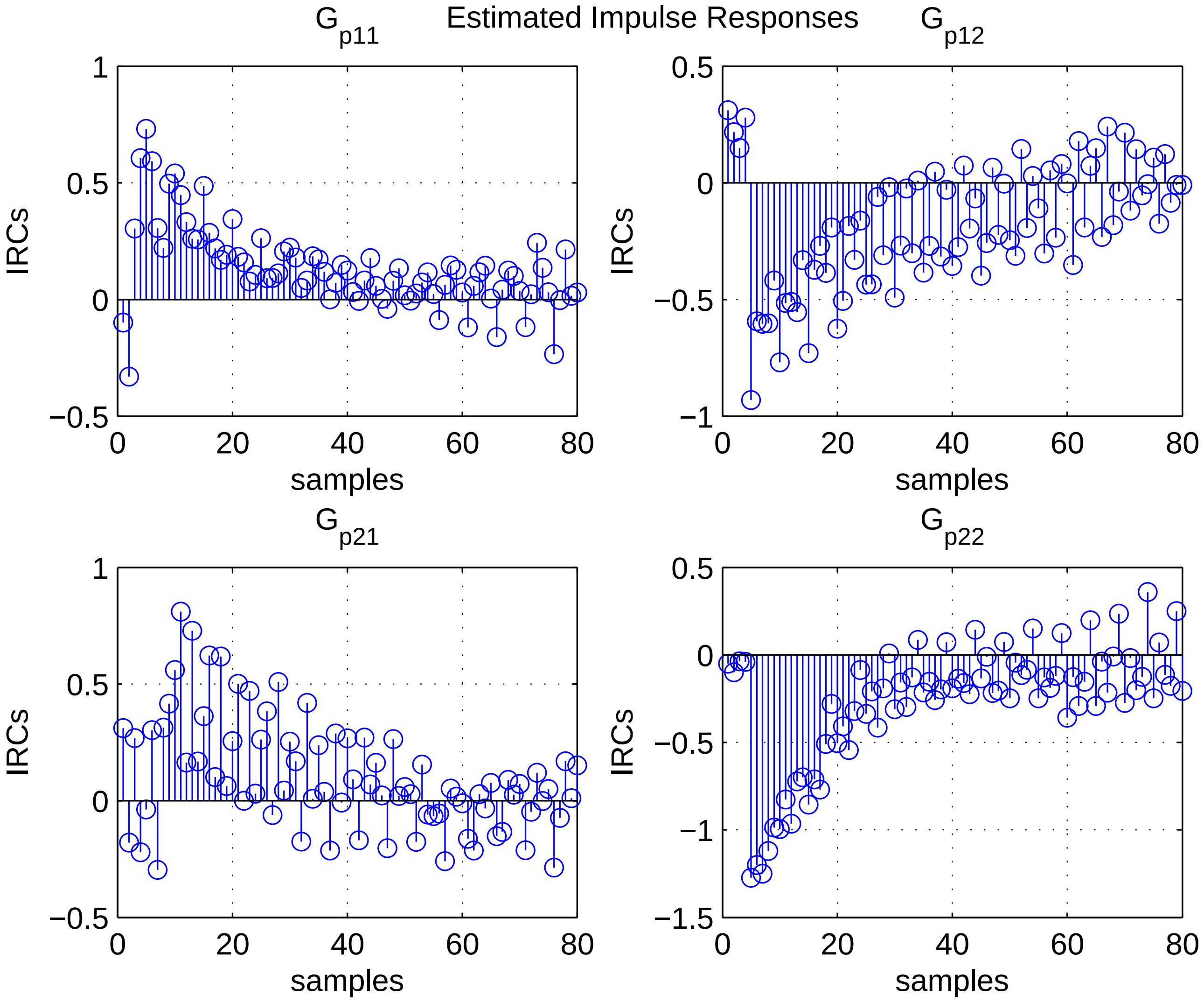

Figure 5.3: Distillation System: Estimated IRCs for each subsystem by Subspace-based method

existing time-delay estimation methods, such as Cumulative Sum (CUSUM) method [74], Separating Frequency Method [75] and Subspace based method [76]. The results show that CUSUM overestimates the time-delay, but Freq.-based method underestimates the time-delay drastically. Time-delay is not detectable (N/D) by using the Subspace-based method. Therefore, the results indicate outstanding performance of the presented RE-based TDE method over the other compared methods.

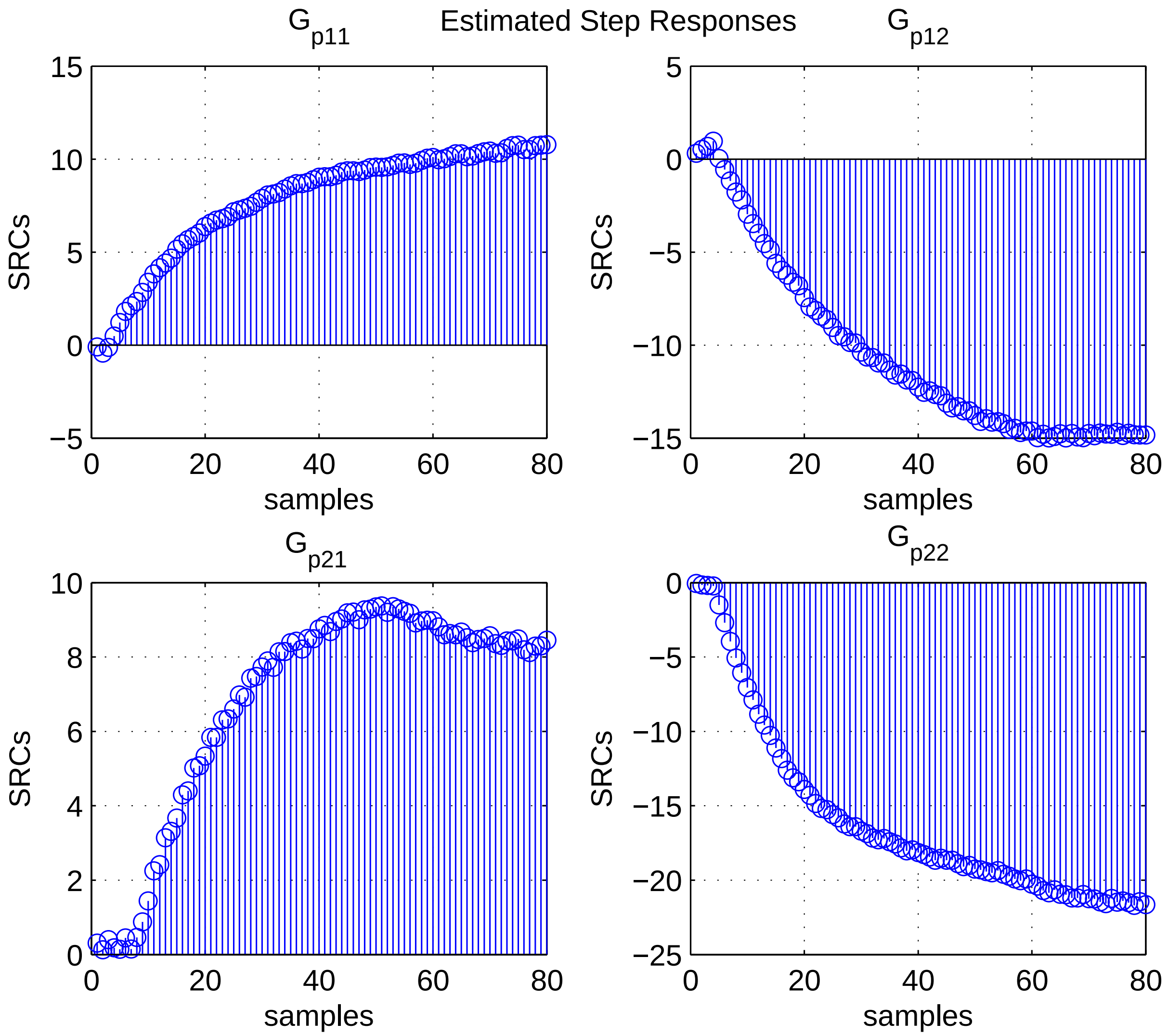

Figure 5.4: Distillation System: Estimated SRCs for each subsystem by Subspace-based method

Table 5.1: Time-delay estimation for System IV from noisy I/O data via different methods

SNR	RE-based	CUSUM	Freq.-based	Subspace-based
0db	11	17	3	N/D
10db	11	15	6	N/D

Distillation System: MIMO time-delayed noisy system

Figure 5.3 and 5.4 show that estimated IRCs and SRCs are highly corrupted due to the additive noise, therefore, time-delay estimation via threshold methods is not a straightforward task. However, by applying the proposed RE-based TDE method the time-delays can be estimated accurately

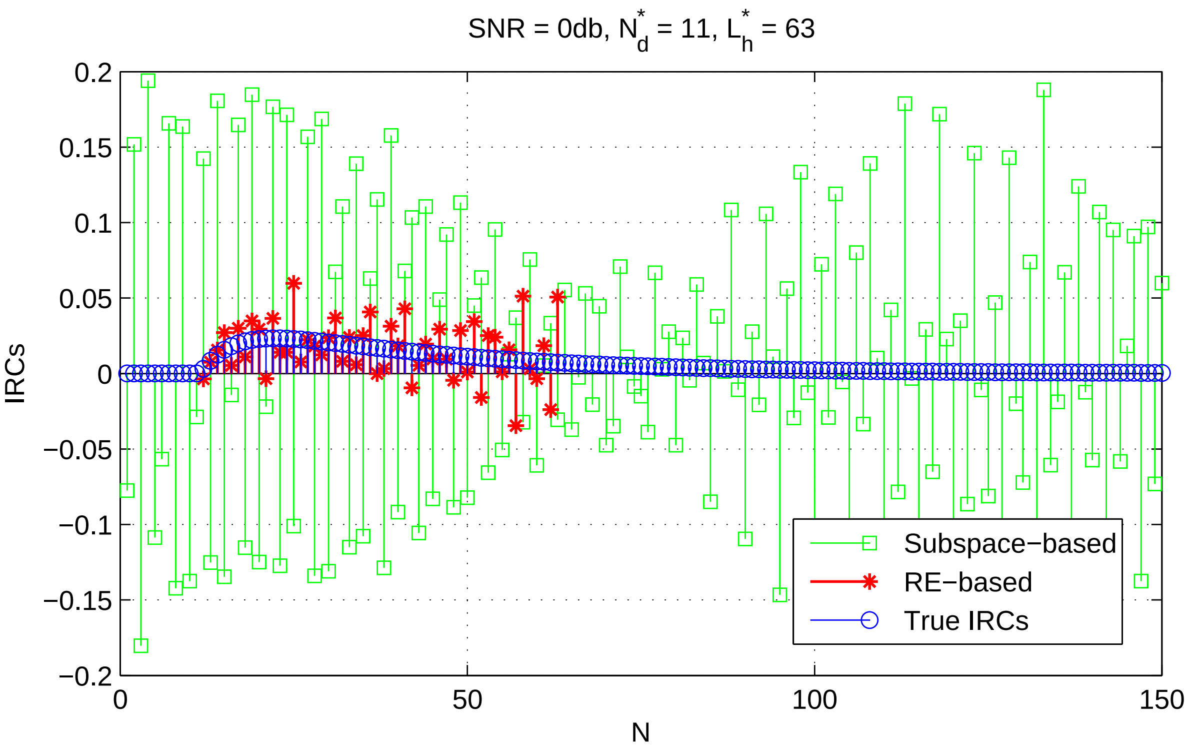

Figure 5.5: System IV: True noise-free IRCs and estimated noisy IRCs for SNR=0db

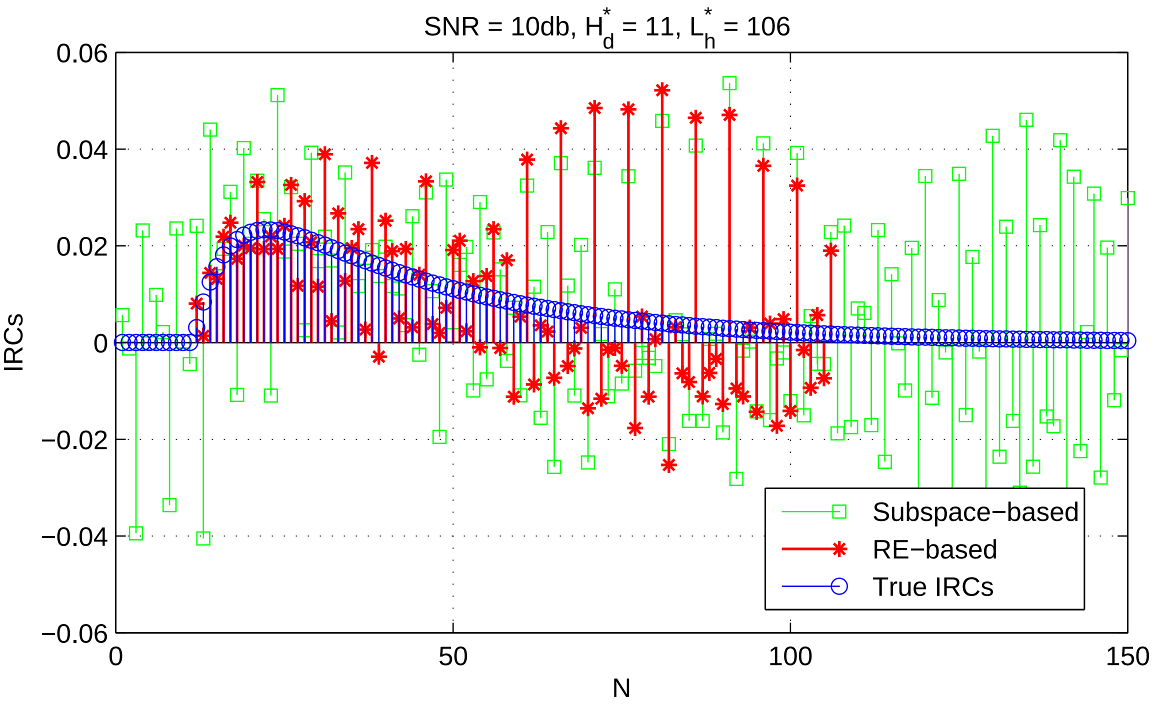

Figure 5.6: System IV: True noise-free IRCs and estimated noisy IRCs for SNR=10db

for each subsystem. Figure 5.7 shows application of RE-based TDE method for this noisy system. RE-based TDE method estimates the time-delay correctly for each subsystem with no need for applying any threshold. Table 5.2 shows estimated sample-delays for each subsystem from noisy data using RE-based TDE method.

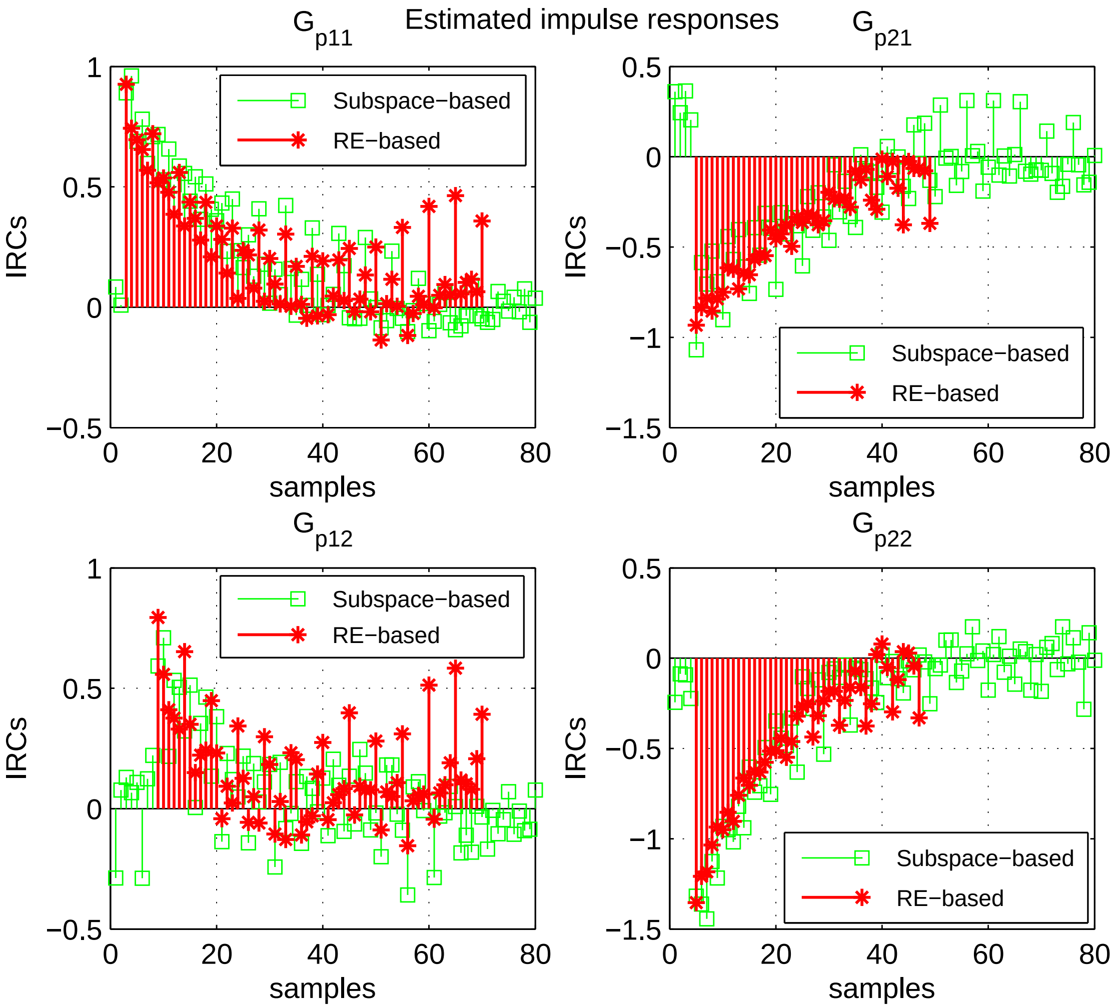

Figure 5.7: Distillation System: Estimated IRCs from noisy I/O data for each subsystem (RE-based TDE method and Subspace-based method)

5.3.4 Order Selection

From the literature review in Section 2.3, in Subspace identification framework the order of the system can be estimated from the dominant singular values of the matrix L_w. Here, drawback of this method is shown in the presence of noisy data, and the application of the presented new MSEE

Table 5.2: Estimated Time-delay for Distillation System using RE-based TDE method

Subsystem	Sample-delay (N_d)
G_{p11}	2
G_{p21}	8
G_{p12}	4
G_{p22}	4

method is illustrated for order estimation.

System IV: SISO time-delayed noisy system

Figure 5.8 shows the first 20 singular values of L_w for noisy I/O data with $SNR = 0$db and $SNR = 10$db, respectively. It can be seen that in noisy data the small singular values of the matrix L_w are magnified, which makes the order estimation difficult by using threshold-based methods.

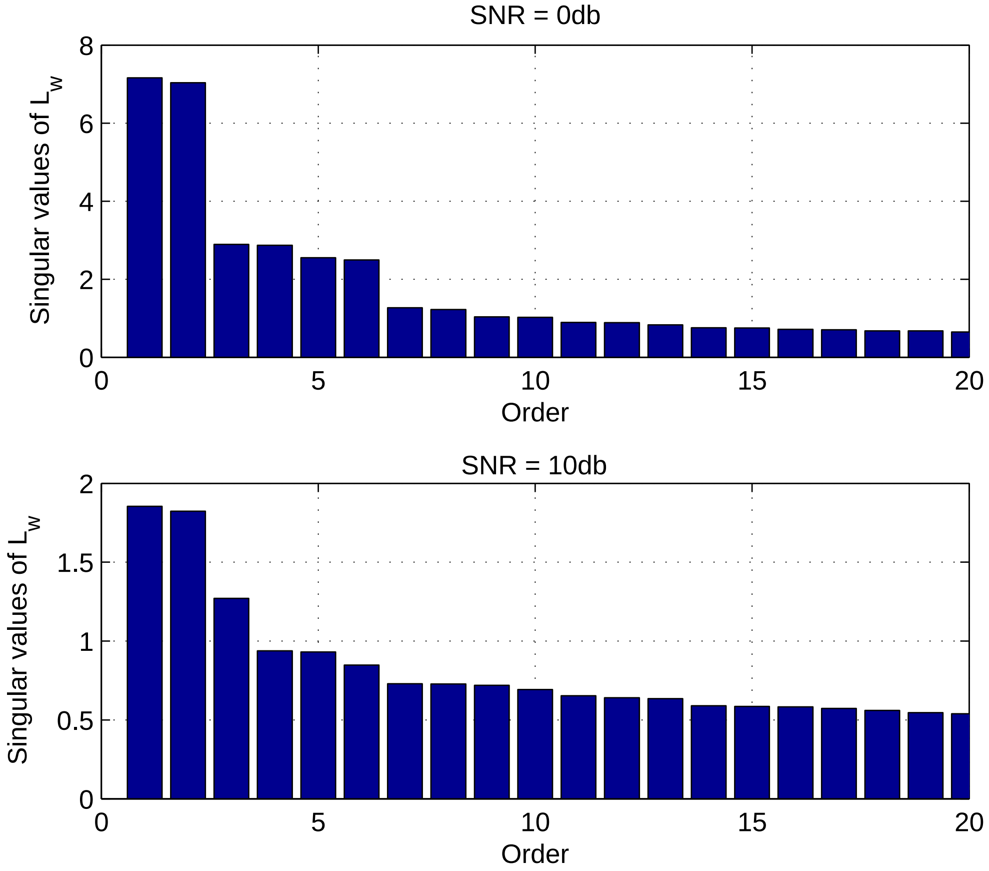

Figure 5.8: System IV: Singular values of L_w from noisy I/O data for SNR=0db and SNR=10db

In these noisy scenarios the order of the system can be estimated accurately by applying the presented MSSE method in Section 5.2.2. Figure 5.9 shows the RE for System IV in $SNR = 0$db. Order selection is performed based on RE minimization by averaging on 100 trials. It is observed that the MSEE method estimates the order precisely and Figure 5.9 indicates $m^* = 2$ for this system for $SNR = 0$db and $SNR = 10$db. Therefore, we have $n = 2$ for System IV.

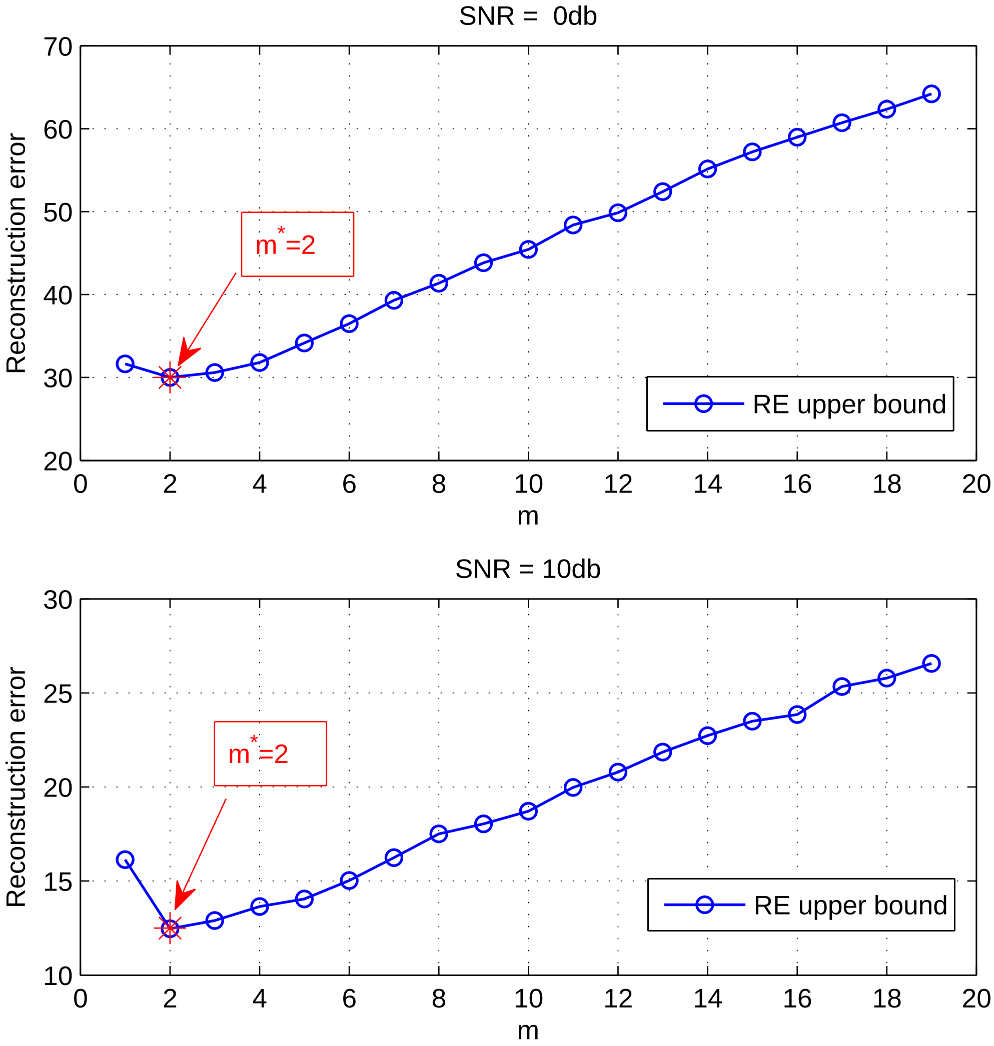

Figure 5.9: System IV: RE upper bound for order selection noisy data with SNR=0db and SNR=10db

In addition, Table 5.3 shows the estimated order for System IV in (5.5) from noisy data using two well-known existing order estimation methods, Akaike's Information Criterion (AIC) [71] and Minimum Description Length (MDL) [72]. The results show that MDL underestimates the order for both SNR values. AIC overestimates for $SNR = 10$db, but by decreasing the SNR value it tends to underestimate the order. The results indicate superiority of presented MSEE method to estimate the system order compared to AIC and MDL.

Table 5.3: Time-delay estimation for System IV from noisy I/O data via different methods

SNR	MSEE	AIC	MDL
0db	2	1.92	1
10db	2	2.67	1.05

Distillation System: MIMO time-delayed noisy system

To estimate the overall order of this MIMO system from noisy data the Subspace-based approach in Section igure 5.10. However, s

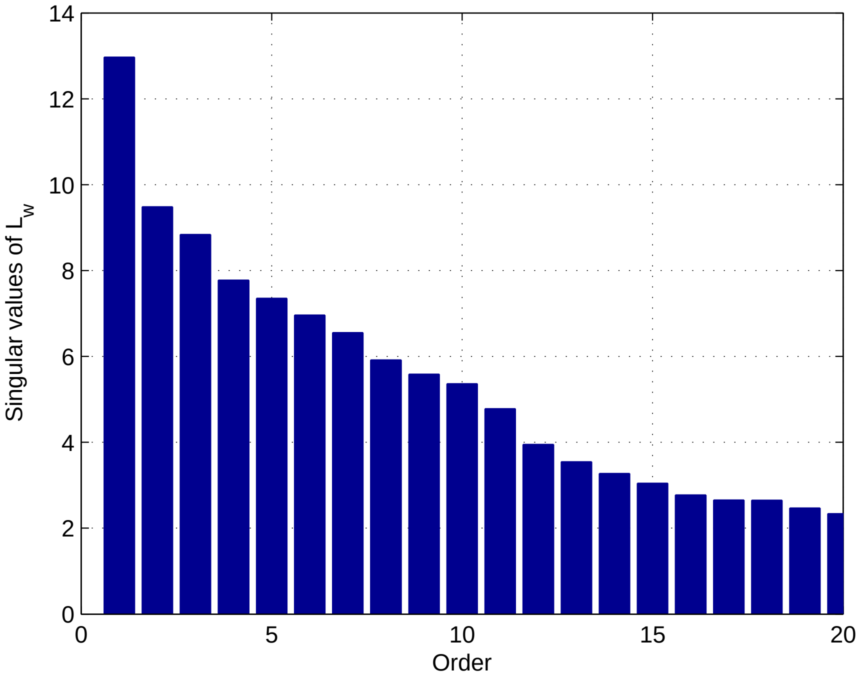

Figure 5.10: Distillation System: Singular values of L_w from noisy data

In noisy cases utilizing the presented MSEE method in Section 5.2.2 as a powerful order selection technique would be helpful. Figure 5.11 shows the order estimation from MSEE method. The results show that the MSEE method can estimate the MIMO system order correctly as $m^* = 1$ from both available noisy outputs y_1 and y_2.

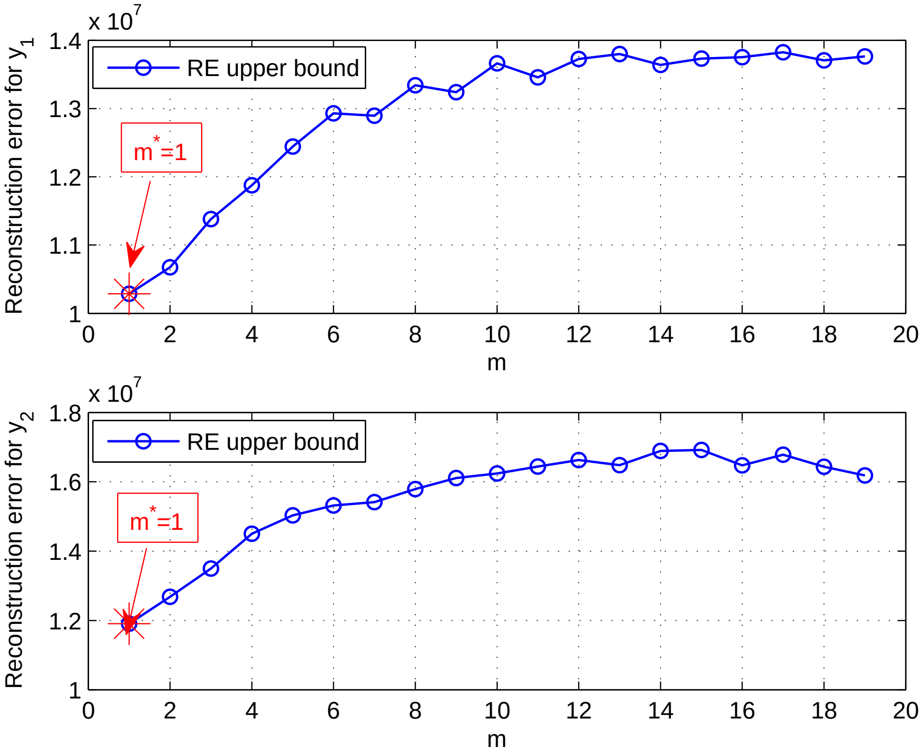

Figure 5.11: Distillation System: RE upper bound for order selection of noisy system

5.4 Summary

In this chapter, a statistical approach was utilized to estimate the time-delay, based on Reconstruction Error (RE) in subspace identification framework. The method is denoted by RE-based TDE. Moreover, application of a new statistical approach was presented to estimate the system order. The method is denoted by MSEE. Comparison of the proposed methods with other classical time-delay estimation and order selection methods illustrates advantages of the proposed methods in different SNR scenarios. The results show efficient performance and robustness of RE-based-TDE and MSEE methods even for very low SNR cases.

Chapter 6

Conclusion and Future Work

Focus of this dissertation was on analysis and improvement of the Subspace Predictive Control (SPC). By closely analyzing the SPC in Chapter 3, its closed-loop transfer function was derived in general form. This enabled us to obtain the necessary and sufficient condition of small-gain SPC closed-loop stability for open-loop stable systems. As a result, SPC stability graph was introduced to find the shortest-prediction-horizon and feasible range (FR) of prediction horizon, which guarantee the closed-loop stability. Next, the efficient control horizon and prediction horizon were derived based on shortest-feasible-prediction-horizon and by minimizing the SPC cost function. Simulation results illustrated the efficiency of the proposed techniques in SPC parameter selection.

On-line adaption of SPC gains was studied in Chapter 4 and the problem of applying persistently exciting (PE) signals in SPC was addressed. This Chapter introduced PSO-based FGS-SPC method to update the SPC gains using FSG rules with no need for applying PE signals. In addition, the proposed PSO-based FGS technique overcame the issue of determining controller gain ranges (CGRs) in FGS method by optimally selecting the CGRs via minimizing the SPC cost function. Simulation results were provided to show the robustness and efficiency of the proposed method.

Chapter 5 discussed black-box data-driven SPC approach in the presence of noisy data. Drawbacks of Subspace-based thresholding approaches for order selection and time-delay estimation were shown in this Chapter. Mean Squared Eigenvalue Error (MSEE) method and Reconstruction Error based time-delay estimation (RE-based TDE) method were implemented for order selection and time-delay estimation, respectively. Simulation results show the superiority of presented MSEE and RE-based TDE methods to other existing approaches.

This dissertation was taking the initiatives to improve the SPC and provide theoretic and systematic requirement for SPC parameter design. There is a great potential in continuing this research direction. For example, extending the SPC stability conditions to open-loop unstable systems, and SPC design by using closed-loop I/O data are interesting possible future research. In addition, it is well worthed to study the design of other parameters of SPC, Q and R, through an optimization approach that is proposed in this dissertation. Lastly, extending the application of PSO-based FGS-SPC method in systems with non-piecewise-constant inputs might be of interest.

Appendix A

Re-configuration of SPC Control Law

Recall SPC control law in (2.150),

$$\Delta u(k+1) = -\tilde{k}_e(y(k) - r(k+1)) - \tilde{k}_{\Delta w_p}\Delta w_p \tag{A.1}$$

and by considering Δw_p in (2.143)

$$\Delta w_p = \begin{bmatrix} \Delta y_p \\ \Delta u_p \end{bmatrix} \tag{A.2}$$

the second part of (A.1) can be rewritten as below,

$$\tilde{k}_{\Delta w_p}\Delta w_p \quad = \quad \begin{bmatrix} \tilde{k}_{\Delta y_p} & \tilde{k}_{\Delta u_p} \end{bmatrix} \begin{bmatrix} \Delta y_p \\ \Delta u_p \end{bmatrix} \tag{A.3}$$

where $\tilde{k}_{\Delta y_p}$ and $\tilde{k}_{\Delta u_p}$ are defined as,

$$\tilde{k}_{\Delta y_p} \quad = \quad \tilde{k}_{\Delta w_p}(:, 1 : Ml) \tag{A.4}$$

$$\tilde{k}_{\Delta u_p} \quad = \quad \tilde{k}_{\Delta w_p}(:, Ml + 1 : M(m + l)) \tag{A.5}$$

Δw_p in (2.143) can be rewritten as follows,

$$\Delta w_p = \begin{bmatrix} \Delta y_p \\ \Delta u_p \end{bmatrix} = \begin{bmatrix} \Delta y(k-M+1) \\ \vdots \\ \Delta y(k-1) \\ \Delta y(k) \\ ---- \\ \Delta u(k-M+1) \\ \vdots \\ \Delta u(k-1) \\ \Delta u(k) \end{bmatrix} = \begin{bmatrix} q^{-M+1}\Delta y(k) \\ \vdots \\ q^{-1}\Delta y(k) \\ \Delta y(k) \\ ---- \\ q^{-M+1}\Delta u(k) \\ \vdots \\ q^{-1}\Delta u(k) \\ \Delta u(k) \end{bmatrix} \tag{A.6}$$

Therefore, (A.3) is obtained as,

$$\begin{aligned} \tilde{k}_{\Delta w_p}\Delta w_p &= \begin{bmatrix} \tilde{k}_{\Delta y_{p_M}} & \dots & \tilde{k}_{\Delta y_{p_2}} & \tilde{k}_{\Delta y_{p_1}} & | & \tilde{k}_{\Delta u_{p_M}} & \dots & \tilde{k}_{\Delta u_{p_2}} & \tilde{k}_{\Delta u_{p_1}} \end{bmatrix} \begin{bmatrix} q^{-M+1}\Delta y(k) \\ \vdots \\ q^{-1}\Delta y(k) \\ \Delta y(k) \\ ---- \\ q^{-M+1}\Delta u(k) \\ \vdots \\ q^{-1}\Delta u(k) \\ \Delta u(k) \end{bmatrix} && \text{(A.7)} \\ &= \tilde{k}_{\Delta y_p}(q^{-1})\Delta y(k) + \tilde{k}_{\Delta u_p}(q^{-1})\Delta u(k) && \text{(A.8)} \end{aligned}$$

where $\tilde{k}_{\Delta y_p}(q^{-1})$ and $\tilde{k}_{\Delta u_p}(q^{-1})$ are defined as,

$$\begin{aligned} \tilde{k}_{\Delta y_p}(q^{-1}) &= \tilde{k}_{\Delta y_{p_M}} q^{-M+1} + \cdots + \tilde{k}_{\Delta y_{p_2}} q^{-1} + \tilde{k}_{\Delta y_{p_1}} && \text{(A.9)} \\ \tilde{k}_{\Delta u_p}(q^{-1}) &= \tilde{k}_{\Delta u_{p_M}} q^{-M+1} + \cdots + \tilde{k}_{\Delta u_{p_2}} q^{-1} + \tilde{k}_{\Delta u_{p_1}} && \text{(A.10)} \end{aligned}$$

As a result, the SPC control law in (A.1) can be rewritten as below,

$$\Delta u(k+1) = -\tilde{k}_e(y(k) - r(k+1)) - \tilde{k}_{\Delta y_p}(q^{-1})\Delta y(k) - \tilde{k}_{\Delta u_p}(q^{-1})\Delta u(k) \tag{A.11}$$

Appendix B

SPC Closed-loop Transfer Function

Recall the open-loop transfer function in (3.1)

$$G_{ol}(q^{-1}) = \frac{y(k)}{u(k)} = \frac{N(q^{-1})}{D(q^{-1})} \tag{B.1}$$

and the SPC control law in (3.2)

$$\Delta u(k+1) = -\tilde{k}_e(y(k) - r(k+1)) - \tilde{k}_{\Delta y_p}(q^{-1})\Delta y(k) - \tilde{k}_{\Delta u_p}(q^{-1})\Delta u(k) \tag{B.2}$$

multiplying both sides of (B.1) by $\Delta = 1 - q^{-1}$ we have,

$$y(k)D(q^{-1}) = u(k)N(q^{-1}) \tag{B.3}$$

$$\Delta y(k)D(q^{-1}) = \Delta u(k)N(q^{-1}) \tag{B.4}$$

$$\Delta y(k)D(q^{-1}) = q^{-1}\Delta u(k+1)N(q^{-1}) \tag{B.5}$$

therefore, $\Delta u(k+1)$ is obtained as follows,

$$\Delta u(k+1) = \Delta y(k)\frac{D(q^{-1})}{q^{-1}N(q^{-1})} \tag{B.6}$$

By replacing (B.6) in (B.2) we have,

$$\Delta y(k)\frac{D(q^{-1})}{q^{-1}N(q^{-1})} = -\tilde{k}_e(y(k) - r(k+1)) - \tilde{k}_{\Delta y_p}(q^{-1})\Delta y(k) - \tilde{k}_{\Delta u_p}(q^{-1})\Delta y(k)\frac{D(q^{-1})}{N(q^{-1})} \tag{B.7}$$

after simplifying, the closed-loop system transfer function is derived as below,

$$G_{cl}(q^{-1}) \quad = \quad \frac{y(k)}{r(k+1)} = \frac{q^{-1}N(q^{-1})}{D_{cl}(q^{-1})} \tag{B.8}$$

where $D_{cl}(q^{-1})$ is the closed-loop characteristics equation as below,

$$D_{cl}(q^{-1}) = \Delta\tilde{k}_e^{-1}D(q^{-1}) + q^{-1}[\Delta\tilde{k}_e^{-1}N(q^{-1})\tilde{k}_{\Delta y_p}(q^{-1}) + \Delta\tilde{k}_e^{-1}D(q^{-1})\tilde{k}_{\Delta u_p}(q^{-1}) + N(q^{-1})] \tag{B.9}$$

Appendix C

Proof of Theorem 3.2.1

Theorem 3.2.1 is proven by applying small-gain root-locus analysis to (3.11). Recall (3.11) as follows,

$$1 + \frac{\tilde{k}_e}{\Delta} \frac{N^*(z^{-1})}{D(z^{-1})} = 0 \tag{C.1}$$

Considering its second part without integrator as below,

$$\tilde{k}_e \frac{N^*(z^{-1})}{D(z^{-1})} \tag{C.2}$$

by having (3.14), DC-gain of the system in (C.2) can be obtained as,

$$\tilde{k}_e \frac{N^*(1)}{D(1)} = \tilde{k}_e \frac{N(1)}{D(1)} = \tilde{k}_e K_{ol} \tag{C.3}$$

Therefore, to utilize the root-locus analysis, it is necessary to know the sign of both $\tilde{k}_e$ and K_{ol}. The system in (C.2) can be rewritten as,

$$\tilde{k}_e \frac{N^*(z^{-1})}{D(z^{-1})} = \tilde{k}_e \frac{(z - z_1)(z - z_2)\ldots(z - z_n)}{(z - p_1)(z - p_2)\ldots(z - p_n)} \tag{C.4}$$

where p_is and z_is are poles and zeros of the compound system $\frac{N^*(z^{-1})}{D(z^{-1})}$. From (C.2) we have,

$$\tilde{k}_e \frac{N^*(1)}{D(1)} = \tilde{k}_e \frac{(1 - z_1)(1 - z_2)\ldots(1 - z_n)}{(1 - p_1)(1 - p_2)\ldots(1 - p_n)} \tag{C.5}$$

Since, open-loop system is stable we have $D(1) > 0$, therefore, the denominator of (C.5) is positive. However, sign of the numerator needs to be determined. For complex conjugated z_is we have,

$$(1 - z_i)(1 - z_i^*) > 0, \quad \text{if} \quad z_i \neq 1 \tag{C.6}$$

for real z_is we have,

$$\begin{cases} 1 - z_i > 0, & \text{if} \quad z_i < 1 \\ 1 - z_i < 0, & \text{if} \quad z_i > 1 \end{cases} \tag{C.7}$$

Therefore, sign of the numerator in (C.5) can be determine as,

$$sgn[\tilde{k}_e(1 - z_1)(1 - z_2)\dots(1 - z_n)] = sgn[\tilde{k}_e(-1)^{N_z}] \tag{C.8}$$

where N_z is the number of real z_is that are greater than one. Therefore, from (C.3) and (C.8) we have,

$$sgn[K_{ol}] = sgn[(-1)^{N_z}] \quad \rightarrow \quad \begin{cases} K_{ol} > 0, & \text{if} \quad N_z \ \ even \\ K_{ol} < 0, & \text{if} \quad N_z \ \ odd \end{cases} \tag{C.9}$$

Rewrite the system with integrator in (3.11) as follows,

$$1 + \frac{\tilde{k}_e z}{z - 1} \frac{(z - z_1)(z - z_2)\dots(z - z_n)}{(z - p_1)(z - p_2)\dots(z - p_n)} = 0 \tag{C.10}$$

For small-gain $\tilde{k}_e$ the roots of (C.10) are close to the open-loop poles $\{p_1, p_2, \dots, p_n\}$. Since, the open-loop poles are stable, therefore, roots of the system in (C.10) are stable for small-gain $\tilde{k}_e \neq 0$.

Here, $z = 1$ is a simple pole and corresponding root-locus follows the real axis at the beginning. Since, the open-loop system is stable, total number of poles and zeros to the right side of $z = 1$ is equal to N_z. Therefore, according to (C.9):

-If N_z is even, then $K_{ol} > 0$ and positive-gain root-locus is plotted, therefore, roots of the SPC close-loop system in (3.11) are inside the unit circle if $\tilde{k}_e > 0$.

-If N_z is odd, then $K_{ol} < 0$ and negative-gain root locus is plotted, therefore, roots of the system in (3.11) are stable if $\tilde{k}_e < 0$.

Therefore, the necessary and sufficient condition for small-gain stability of SPC closed-loop system is,

$$\tilde{k}_e K_{ol} > 0 \tag{C.11}$$

Bibliography

[1] D.C. Karnopp and D.L. Margolis. *Engineering Applications of Dynamics*. John Wiley & Sons, 2008.

[2] B.T. Kulakowski, J.F. Gardner, and J.L. Shearer. *Dynamic Modeling and Control of Engineering Systems*. Cambridge University Press, 2007.

[3] M.J. Moran and H.N. Shapiro. *Fundamentals of Engineering Thermodynamics*. Wiley, 2008.

[4] L. Ljung. *System Identification: Theory for User*. Prentice Hall, NJ, 1999.

[5] T.P. Bohlin. *Practical Grey-Box Process Identification: Theory and Application*. Springer, 2006.

[6] C.E. Garcia, D.M. Prett, and M. Morari. Model predictive control: Theory and practice; a survey. *Automatica*, 25:335–348, 1989.

[7] J.H. Lee. Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9(3):415–424, 2011.

[8] D.Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50:2967–2986, 2014.

[9] M. Morari and J.H. Lee. Model predictive control: Past, present and future. *Computers and Chemical Engineering*, 23(4):667–682, 1999.

[10] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[11] J.M. Maciejowski. *Predictive Control with Constraints*. Pearson Education Limited, 2002.

[12] L. Wang. *Model Predictive Control System Design and Implementation using MATLAB.* Springer-Verlag, 2009.

[13] C.R. Cutler and B.L. Ramaker. Dynamic matrix control - a computer control algorithm. *AIChE 86th National Meeting, Texas*, 1979.

[14] C.E. Garcia and A.M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Community*, 46:073–087, 1986.

[15] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control - Part I. the basic algorithm. *Automatica*, 23(2):137–148, 1987.

[16] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control - Part II. extensions and interpretations. *Automatica*, 23(2):149–160, 1987.

[17] J.B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20:38–52, 2000.

[18] W. Favoreel and B. De Moor. SPC: Subspace predictive control. *Proceeding of the 14th IFAC World Congress [CD-ROM]*, H:235–240, 1999.

[19] T. Katayama. *Subspace Methods for System Identification.* Springer-Verlag, London, 2005.

[20] P.V. Overschee and B. De Moor. *Subspace Identification of Linear Systems: Theory, Implementation, Application.* Kluwer Academic Publisher, 1996.

[21] R. Kadali, B. Huang, and A. Rossiter. A data driven subspace approach to predictive controller design. *Control Engineering Practice*, 11(3):535–561, 2003.

[22] S.T. Navalkar, E.V. Solingen, and J.W.V. Wingerden. Wind tunnel testing of subspace predictive repetitive control for variable pitch wind turbines. *IEEE Transactions on Control Systems Technology*, 23(6):2101–2116, 2015.

[23] J. Richalet. Industrial applications of model predictive based predictive control. *Automatica*, 29(5):1251–1274, 1993.

[24] A.S. Kumar and Z. Ahmad. Model predictive control and its current issues in chemical engineering. *Chemical Engineering Communication*, 199(4):472–511, 2012.

[25] J. Zeng, C. Gao, and H. Su. Data-driven predictive control for blast furnace ironmaking process. *Computers and Chemical Engineering*, 34:1854–1862, 2010.

[26] S. Jing, R. Errouissi, A. Al-Durra, and I. Boiko. A data-driven subspace predictive controller design for artificial gas-lift process. *IEEE Conference on Control Application (CCA)*, pages 1179–1184, 2015.

[27] G. Valverde and T. Van Cutsem. Model predictive control of voltages in active distribution networks. *IEEE Transactions on Smart Grid*, 4(4):2152–2161, 2013.

[28] X. Wu, J. Shen, Y. Li, and K.Y. Lee. Data-driven modeling and predictive control for boiler-turbine unit. *IEEE Transaction on Energy Conversion*, 28(3):470–481, 2013.

[29] X. Luo. Data-driven predictive control for continuous-time linear parameters varying systems with application to wind turbine. *International Journal of Control, Automation and Systems*, 15(2):619–626, 2017.

[30] J. Hu and P. Karava. Model predictive control strategies for buildings with mixed-mode cooling. *Building and Environment*, 71:233–244, 2014.

[31] S.E. Shafiei, T. Knudsen, R. Wisniewski, and P. Andersen. Data-driven predictive direct load control of refrigeration systems. *IET Control Theory and Applications*, 9(7):1022–1033, 2015.

[32] Y. Xia, W. Xie, B. Liu, and X. Wang. Data-driven predictive control for network control systems. *Information Science*, 235:45–54, 2013.

[33] P. Wang, D. Zhu, and X. Lu. Active queue management algorithm based on data-driven predictive control. *Telecommunication Systems*, 64(1):103–111, 2017.

[34] X. Lu, H. Chen, B. Gao, Z. Zhang, and W. Jin. Data-driven predictive gearshift control for dual-clutch transmissions and FPGA implementation. *IEEE Transactions on Industrial Electronics*, 62(1):599–610, 2015.

[35] B. Kulcsar, J.W. wan Wingerden, J. Dong, and M. Verhaegen. Closed-loop subspace predictive control for hammerstein systems. *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 2604–2609, 2009.

[36] W. Lu, L. Ning, and L. Shao-Yuan. Performance monitoring of the data-driven subspace predictive control systems based on historical objective function benchmark. *ACTA Automatica SINICA*, 39(5):542–547, 2013.

[37] D. Mayne and P. Falugi. Generalized stabilizing conditions for model predictive control. *Journal of Optimization Theory and Application*, 169:719–734, 2016.

[38] C. Panjapornpon and M. Soroush. Shortest-prediction-horizon non-linear model-predictive control with guaranteed asymptotic stability. *International Journal of Control*, 80(10):1533–1543, 2007.

[39] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 30:789–814, 2000.

[40] D. Viudez-Morerias. Extended predictive control: Stability and performance. *Journal of Dynamic Systems, Measurement and Control*, 137, 2015.

[41] R. Estrada, A. Favela, A. Raimondi, and A. Nevado. Stable predictive control horizons. *International Journal of Control*, 85(4):361–372, 2012.

[42] J.A. Rossiter, J.R. Gossner, and B. Kouvaritakis. Infinite horizon stable predictive control. *IEEE Transactions on Automatic Control*, 41(10), 1996.

[43] P.O.M. Scokaert and D.W. Clarke. Stabilizing properties of constrained predictive control. *IEE Proceeding on Control Theory and Application*, 141(5):295–304, 1994.

[44] K. Patan. Neural network-based model predictive control: Fault tolerance and stability. *IEEE Transactions on Control Systems Technology*, 23(3):1147–1155, 2015.

[45] R. Scattolini and S. Bittanti. On the choice of the horizon in long-range predictive control-some simple criteria. *Automatica*, 26(5):915–917, 1990.

[46] J.O. Trierweiler and L.A. Farina. RPN tuning strategy for model predictive control. *Journal of Process Control*, 13:591–598, 2003.

[47] A. Georgiou, C. Georgakis, and W.L. Luyben. Nonlinear dynamic matrix control for high-purity distillation columns. *AIChE Journal*, 34(8):1287–1298, 1988.

[48] J.L. Garriga and M. Soroush. Model predictive controller tuning methods: A review. *Industrial Engineering Chemical Research*, 49:3505–3515, 2010.

[49] P. Bagheri and A. Khaki-Sedigh. An analytical tuning approach to multivariable model predictive controllers. *Journal of Process Control*, 24:41–54, 2014.

[50] J. Yu and S.J. Qin. Statistical MIMO controller performance monitoring. Part I: data-driven covariance benchmark. *Journal of Process Control*, 18:277–296, 2008.

[51] J. Yu and S.J. Qin. Statistical MIMO controller performance monitoring. Part II: performance diagnosis. *Journal of Process Control*, 18:297–319, 2008.

[52] Q. Zhang and S. Li. Enhanced performance assessment of subspace model-based predictive controller with parameters tuning. *The Canadian Journal of Chemical Engineering*, 85:537–548, Aug. 2007.

[53] X. Tian, G. Chen, and S. Chen. A data-based approach for multivariate model predictive control performance monitoring. *Neurocomputing*, 74:588–597, 2011.

[54] R. Shridhar and D.J. Cooper. A tuning strategy for unconstrained multi-variable model predictive control. *Industrial Engineering Chemical Res.*, 37(10):4003–4016, 1998.

[55] R. Shridhar and D.J. Cooper. A tuning strategy for unconstrained SISO model predictive control. *Industrial Engineering Chemical Res.*, 36(3):729–746, 1997.

[56] J.L. Garriga and M. Soroush. Model predictive controller tuning via eigenvalue placement. *American Control Conference*, pages 429–434, 2008.

[57] A.S. Yamashita, A.C. Zanin, and D. Odloak. Tuning the model predictive control of a crude distillation unit. *ISA Transactions*, 60:178–190, 2016.

[58] K.Y. Rani and H. Unbehauen. Study of predictive controller tuning methods. *Journal of Process Control*, 33(12):2243–2248, 1997.

[59] C. Muller, D.E. Quevedo, and G.C. Goodwin. How good is quantized model predictive control with horizon one? *IEEE Transaction on Automatic Control*, 56(11):2623–2638, 2011.

[60] S. Valluri, M. Soroush, and M. Nikravesh. Shortest-prediction-horizon non-linear model-predictive control. *Chemical Engineering Science*, 53(2):273–292, 1998.

[61] J. Dong and M. Verhaegen. On the equivalence of closed-loop subspace predictive control with LQG. *Proceeding of the 47th IEEE Conference on Decision and Control*, pages 4085–4090, 2008.

[62] W. Favoreel, B. De Moor, M. Gevers, and P. Van Overschee. Closed-loop model-free subspace-based LQG-design. *Proceeding of the 7th Mediterranean Conf. on Control and Automation (MED99)*, pages 1926–1939, June, 1999.

[63] E. Zacokova, S. Privara, and M. Pcolka. Persistent excitation condition within the dual control framework. *Journal of Process Control*, 23:1270–1280, 2013.

[64] E. Aggelogiannaki and H. Sarimveis. Multiobjective constrained MPC with simultaneous closed-loop identification. *International Journal of Adaptive Control and Signal Processing*, 20(4):145–173, 2006.

[65] G. Marafioti, R.R. Bitmead, and M. Hovd. Persistently exciting model predictive control. *International Journal of Adaptive Control and Signal Processing*, 28(6):536–552, 2014.

[66] R. Hallouzi and M. Verhaegen. Fault-tolerant subspace predictive control applied to a boeing 747 model. *Journal of Guidance, Control and Dynamics*, 31(4):873–883, 2008.

[67] R. Hallouzi and M. Verhaegen. Reconfigurable fault tolerant control of a boeing 747 using subspace predictive control. *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, USA*, 2007.

[68] N.A. Mardi and L. Wang. Subspace-based model predictive control of time-varying systems. *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Conference*, pages 4005–4010, 2009.

[69] L. Zhang, S.Z. Xu, and H.T. Zhao. Adaptive subspace predictive control with time-varying forgetting factor. *International Journal of Automation and Computing*, 11(2):205–209, 2014.

[70] R. Hallouzi and M. Verhaegen. Persistency of excitation in subspace predictive control. *Proceedings of the 17th World Congress IFAC*, 2008.

[71] H. Akaike. Information theory and an extension of the maximum likelihood principle. *Proceeding of the Second International Symposium Information Theory*, pages 267–281, 1973.

[72] J. Rissanen. Universal coding, information, prediction and estimation. *IEEE Transactions on Information Theory*, 30:629–636, 1984.

[73] S. Kritchman and B. Nadler. Non-parametric detection of number of signals: Hypothesis testing and random matrix theory. *IEEE Transactions on Signal Processing*, 57(10):3930–3941, 2009.

[74] F. Gustaffson. *Adaptive Filtering and Change Detection.* Wiley, West Sussex, England, 2000.

[75] S. Bjorklund and L. Ljung. A review of time-delay estimation techniques. *American Control Conference (ACC), Washington, USA*, pages 95–100, 2013.

[76] J. Shalchian, A. Khaki-Sedigh, and A. Fatehi. A subspace based method for time delay estimation. *Proceeding of the 4th International Symposium on Communication, Control and Signal Processing (ISCCSP), Cyprus*, 2010.

[77] N.K. Sinha and B. Kuszta. *Modeling and Identification of Dynamic Systems.* Van Nostrand Reinhold Company, USA, 1983.

[78] T. Soderstrom and P. Stoica. *System Identification.* Prentice Hall, London, 1989.

[79] K. Godfrey. *Perturbation Signals for System Identification.* Prentice Hall International, New York, 1993.

[80] K. J. Astrom and B. Wittenmark. *Computer Controlled Systems.* Prentice Hall, New Jersey, 1984.

[81] K. J. Astrom and T. Bohlin. Numerical identification of linear dynamic systems from normal operating records. *IFAC Symposium on Self-Adaptive Systems, Teddington, England*, 1965.

[82] G.E. Box and G.M. Jenkins. *Time Series Analysis, Forecasting and Control.* Holden-Day, San Francisco, CA, 1970.

[83] P. V. Kabaila and G. C. Goodwin. On the estimation of the parameters of an optimal interpolator when the class of interpolators is restricted. *SIAM Journal on Control and Optimization*, 18(2):121–144, 1980.

[84] P.E. Wellstead. Non-parametric methods of system identification. *Automatica*, 17:55–69, 1981.

[85] H. Rake. Step response and frequency response methods. *Automatica*, 16:519–526, 1980.

[86] K. R. Godfrey. Correlation methods. *Automatica*, 16:527–534, 1980.

[87] L. Ljung. Prediction error estimation methods. *Circuits, Systems and Signal Processing*, 21(1):11–21, 2002.

[88] T. Soderstrom and P. Stoica. Comparison of some instrumental variable methods: Consistency and accuracy aspects. *Automatica*, 17:101–115, 1981.

[89] T. Soderstrom and P. Stoica. *Instrumental Variable Methods for System Identification.* Springer-Verlag, New York, 1983.

[90] M. Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12):1835–1851, 1995.

[91] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[92] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[93] S. Beheshti and M.A. Dahleh. LTI systems, additive noise and order estimation. *IEEE Conference on Decision and Control, Hawaii USA*, 2003.

[94] S. Beheshti and M.A. Dahleh. A new information-theoretic approach to signal denoising and best basis selection. *IEEE Transactions on Signal Processing*, 53(10):3613–3624, 2005.

[95] R. E. Kalman and B. L. Ho. Effective construction of linear state-variable models from input/output functions. *Regelungstechnik*, 14(12):545–548, 1966.

[96] H. Akaike. Stochastic theory of minimal realization. *IEEE Transactions on Automatic Control*, AC-19(6):667–674, 1974.

[97] P.V. Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994.

[98] B. Huang and R. Kadali. *Dynamic Modeling, Predictive Control and Performance Monitoring - A Data-driven Subspace Approach.* Springer, 2008.

[99] K. J. Astrom and B. Wittenmark. *Adaptive Control.* Addison-Wesley Longman Publishing, Boston, MA, USA, 1994.

[100] X. Luo and Y. Song. Adaptive predictive control: A data-driven closed-loop subspace identification approach. *Abstract and Applied Analysis*, 2014.

[101] M. Verhaegen and P. Dewilde. Subspace model identification. Part I: the output-error state-space model identification class of algorithms. *International Journal of Control*, 56(5):1187–1210, 1992.

[102] M. Verhaegen. Application of a subspace model identification technique to identify LTI systems operating in closed loop. *Automatica*, 29(4):1027–1040, 1993.

[103] E. Fernandez-Camach and C.Bordons-Alba. Model predictive control in the process industry. 1995.

[104] J. Richalet and D. O'Donovan. *Predictive Functional Control: Principles and Industrial Applications*. 2009.

[105] D.J. Lamburn, P.W. Gibbens, and S.J. Dumble. Efficient constrained model predictive control. *European Journal of Control*, 20:301–311, 2014.

[106] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovations form from input-output data. *Automatica, Special Issue on Statistical Signal Processing and Control*, 30(1):61–74, 1994.

[107] C.S. Chiu, K.Y. Lian, and P. Liu. Fuzzy gain scheduling for parallel parking a car-like robot. *IEEE Transactions on Control Systems Technology*, 13(6):1084–1092, 2005.

[108] H. Kakigano, Y. Miura, and T. Ise. Distribution voltage control for DC micro-grids using fuzzy control and gain scheduling technique. *IEEE Transactions on Power Electronics*, 28(5):2246–2258, 2013.

[109] P. Sarma. Multivariable gain-scheduled fuzzy logic control of an exothermic reactor. *Engineering Applications of Artificial Intelligence*, 14:457–471, 2001.

[110] H. Yang, X. Li, Z. Liu, and L. Zhao. Robust fuzzy gain scheduling control for nonlinear systems subject to actuator saturation via delta operator approach. *International Journal of Information Science*, 272(C):158–172, 2014.

[111] K. Bedoud, M. Alirachedi, T. Bahi, and R. Lakel. Adaptive fuzzy gain scheduling of PI controller for control of the wind energy conversion systems. *Energy Procedia*, 74:211–225, 2015.

[112] C.C. Huang and W.H. Yu. Applying a fuzzy gain-scheduled PID controller to dyebath ph. *Textile Research Journal*, 71(12):1074–1078, Dec. 2001.

[113] Y. Kanthaphayao and V. Chunkag. Current-sharing bus and fuzzy gain scheduling of proportional-integral controller to control a parallel-connected AC/DC converter. *IET Power Electronic*, 7(10):2525–2532, 2014.

[114] A. Rodriguez-Martinez, R. Garduno-Ramirez, and L.G. Vela-Valdes. PI fuzzy gain-scheduling speed control at startup of gas-turbine power plant. *IEEE Transactions on Energy Conversion*, 26(1):310–317, 2011.

[115] T. Chaiyatham and I. Ngamroo. Alleviation of power fluctuation in a microgrid by electrolyzer based on optimal fuzzy gain scheduling PID control. *IEEJ Transactions on Electrical and Electronic Engineering*, 9(2):158–164, 2014.

[116] Z.W. Woo, H.Y. Chung, and J.J. Lin. A PID type fuzzy controller with self-tuning scaling factors. *Fuzzy Sets and Systems*, 115(2):321–326, 2000.

[117] Z.Y. Zhao, M. Tomizuka, and S. Isaka. Fuzzy gain scheduling of PID controllers. *IEEE Transactions on Systems, Man and Cybernetics*, 23(5):1392–1398, 1993.

[118] S. Bouallegue, J. Haggege, M. Ayadi, and M. Benrejeb. PID-type fuzzy logic controller tuning based on particle swarm optimization. *Engineering Application of Artificial Intelligence*, 25:484–493, 2012.

[119] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pages 39–43, 1995.

[120] M. Clerc and J. Kennedy. The particle swarm - explosion, stability, and, convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[121] Y. Shi and R. Eberhart. A modified particle swarm optimizer. *IEEE International Conference on Evolutionary Computation*, pages 69–73, 1998.

[122] C. Blum and D. Merkle. *Swarm Intelligence, Introduction and Applications.* Springer, Berlin, Germany, 2008.

[123] S.H. Ling, K.Y. Chan, F.H.F. Leung, F. Jiang, and H. Nguyen. Quality and robustness improvement for real world industrial systems using fuzzy particle swarm optimization. *Engineering Applications of Artificial Intelligence*, 47:68–80, 2016.

[124] Y. Marinakis, M. Marinaki, and G. Dounias. A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 23(3):463–472, 2010.

[125] A.P. Engelbrecht. *Computational Intelligence: An Introduction, 2nd Edition.* John Wiley and Sons, 2007.

[126] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8:204–210, 2004.

[127] P. R. Belanger. On type 1 systems and the Clarke-Gawthrop regulator. *Automatica*, 19(1):91–94, 1983.

[128] D.W. Clarke, M.A. D.Phil, and P.J. Gawthrop. Self-tuning controller. *IEE Proceedings of the Institution of Electrical Engineering*, 122(9):929–934, 1975.

[129] S. Sedghizadeh and S. Beheshti. Stability and horizons tuning of subspace predictive control. *submitted to: Automatica*, 2017.

[130] W. Favoreel, B. De Moor, M. Gevers, and P. Van Overschee. Model-free subspace-based LQG-design. *Proceeding of the American Control Conference*, pages 3372–3376, June, 1999.

[131] L.Y. Li, T.T. Dong, S. Zhang, X.X. Zhang, and S.P. Yang. Time-delay identification in dynamic processes with disturbance via correlation analysis. *Control Engineering Practice*, 62:92–101, 2017.

[132] V.D. Hajare, A.A. Khandekar, and B.M. Patre. Discrete sliding mode controller with reaching phase elimination for TITO systems. *ISA Transactions*, 66:32–45, 2017.

[133] R.K. Wood and M.W. Berry. Terminal composition control of a binary distillation column. *Chemical Engineering Science*, 28(9):1707–1717, 1973.

[134] R. Hallouzi and M. Verhaegen. Subspace predictive control applied to fault-tolerant control. *Fault Tolerant Flight Control, Springer-Verlag Berlin Heidelberg, LNCIS 399*, pages 293–317, 2010.

[135] S. Sedghizadeh and S. Beheshti. Fuzzy gain scheduling of subspace predictive controller. *American Control Conference*, pages 2693–2698, 2016.

[136] E. Poulin, A. Pomerleau, A. Desbiens, and D. Hodouin. Development and evaluation of an auto-tuning and adaptive PID controller. *Automatica*, 32(1):71–82, 1996.

[137] I. H. Altas. A fuzzy logic controlled tracking system for moving targets. *Proceeding of the 112th IEEE International Symposium on Intelligent Control*, pages 43–48, Jul. 1997.

[138] Z. Muhammad, Z.M. Yusoff, N. Kasuan, M.N.N. Nordin, M.H.F. Rahiman, and M.N. Taib. Online tuning PID using fuzzy logic controller with self-tuning method. *IEEE 3rd International Conference on System Engineering and Technology*, pages 94–98, 2013.

[139] A. Visioli. Tuning of PID controllers with fuzzy logic. *IEEE Proceedings on Control Theory Application*, 148(1):1–8, 2001.

[140] S. Sedghizadeh and S. Beheshti. Particle swarm optimization based fuzzy gain scheduled subspace predictive control. *submitted to: Engineering Applications of Artificial Intelligence*, 2017.

[141] L. Ljung. System identification toolbox for use with MATLAB. 1991.

[142] S. Beheshti, S. Sedghizadeh, and A. Sahebalam. Reconstruction error (RE) based time delay estimation. *submitted to: IEEE Signal Processing Letters*, 2017.

[143] S. Beheshti and S. Sedghizadeh. Number of source signal estimation by mean squared eigenvalue error (MSEE). *submitted to: IEEE Transactions on Signal Processing*, 2017.

[144] S. Beheshti and M.A. Dahleh. Noisy data and impulse response estimation. *IEEE Transactions on Signal Processing*, 58(2):510–521, 2010.

[145] S. Beheshti, M. Hashemi, E. Sejdic, and T. Chau. Mean square error estimation in thresholding. *IEEE Signal Processing Letters*, 18(2), 2011.