Sampled-Data Generalized Predictive Control (SDGPC)

by Guoqiang Lu

B.Sc. Beijing Institute of Technology, 1984

M.Sc. Beijing Institute of Technology, 1987

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

January 1996

© Guoqiang Lu, 1996

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver, Canada

Date Feb. 8th, 1996

DE-6 (2/88)

Abstract

This thesis develops a novel predictive control strategy called Sampled-Data Generalized Predictive Control (SDGPC). SDGPC is based on a continuous-time model yet assumes the projected control profile to be piecewise constant, i.e. to be compatible with zero order hold circuit. It thus enjoys both the advantage of continuous-time modeling and the flexibility of digital implementation. SDGPC is shown to be equivalent to an infinite horizon LQ control law under certain conditions. For well-damped open-loop stable systems, the piecewise constant projected control scenario adopted in SDGPC is shown to have benefits such as reduced computational burden, increased numerical robustness etc. When extending SDGPC to tracking design, it is shown that future knowledge of the setpoint significantly improves tracking performance. A two-degree-of-freedom SDGPC based on optimization of two performance indices is proposed. Actuator constraints are considered in an anti-windup framework. It is shown that the nonlinear control problem is equivalent to a linear time-varying problem. The proposed anti-windup algorithm is also shown to have attractive stability properties. Time-delay systems are treated later. It is shown that the Laguerre-filter-based adaptive SDGPC has excellent performance controlling systems with varying time-delay. An algorithm for continuous-time system parameter estimation based on sampled input output data is presented. The effectiveness and the advantages of continuous-time model estimation and the SDGPC algorithm over the pure discrete-time approach are highlighted by an inverted pendulum experiment.

Table of Contents

List of Tables

List of Figures

Acknowledgment

I would like to express my deep gratitude to Professor Guy A. Dumont for giving me the opportunity to pursue a PhD degree in process control under his supervision. I wish to thank him for his invaluable guidance and instruction throughout my study in UBC. I thank him for helping me selecting a project for his self-tuning control course which led to the research reported in this thesis. I would also like to thank Professor Michael S. Davies, Dr. Ye Fu and other members of the Control Group at the Pulp and Paper Centre for their helping hands. I also appreciate the constructive suggestions and criticisms of the members of my examination committee for making the draft more readable.

Financial support from Professor Guy A. Dumont and the Woodpulp Network of Centres of Excellence is greatly acknowledged.

I would like to thank my parents, brother and sister for their love and care. They were always there especially during the hard times.

I am grateful to my wife Weiling, who always believed in me and offered unlimited support. I would like to thank her for her love and patience during those years.

Chapter 1

Introduction

1.1 Background and Motivation

Model Based Predictive Control (MBPC) has achieved a significant level of success in industrial applications during the last ten years. This has inspired the academic community to investigate the theoretical foundations of MBPC. As a result, a wealth of exciting stability results have been obtained for the last couple of years. It is safe to say that a solid theoretical foundation for model predictive control has now been established.

One of the many explanations of the success of MBPC is that predictive control is an open methodology. That is, within the framework of predictive control, the predictive controller can be closely tailored to meet different requirements of a particular problem. As a result, quite a few predictive controllers have been proposed. Some of the well-known predictive controllers are GPC (Generalized Predictive Control [13]), DMC (Dynamic Matrix Control [15]), Model Predictive Heuristic Control [61], etc. All of these controllers are developed in a discrete-time context. That is, all the controller designs start with a discrete-time model which can be obtained either by direct identification from the discrete input output data or by discretizing a continuous-time model. Although most of the industrial processes are continuous in nature, the discrete-time approach of MBPC is a natural choice since most of the MBPC algorithms need computer implementation. However, the selection of the sampling interval in digital control is not a trivial task. Moreover, it has been pointed out that in applications where fast sampling is needed, the discrete-time model in z-domain is not a good description of the underlying continuous-time process since the poles and zeros of the continuous-time system are mapped to the unit circle as the sampling interval Δ goes to zero. It is thus not a surprise to see a resurgence of interest in continuous-time model based methods [33] [32]. Efforts have also recently been made to unify discrete-time and continuous-time methods under the name of δ-operator [50].

One of the few continuous-time results on MBPC is the work done by H.Demircioglu and P.J.Gawthrop [18] in which Continuous-time Generalized Predictive Control (CGPC) was developed based on Laplace transfer function model. Multivariable CGPC [19] and modified CGPC with guaranteed stability are also available [16]. However, results on continuous-time MBPC are still very limited compared with its discrete-time counterpart. There is still no reported real life application of continuous-time MBPC to the best of the author's knowledge. This is perhaps partly due to the fact that it assumes the projected future control inputs to be of a polynomial type which is not compatible with the widely used zero-order hold device in digital control equipment. As a result, the digital implementation of CGPC unavoidably introduces approximations which often demand a small sampling interval. This demand will result in computation difficulties in some applications. Nonetheless, continuous-time modelling is still appealing even for the purpose of digital control since physical relevance of the model parameters is retained and it is easier to identify partially-known systems in a continuous-time setting. This motivates us to develop a MBPC algorithm based on continuous-time modelling while assuming the projected future control scenario to be piecewise constant, i.e. to be compatible with the zero-order hold device. The model form is chosen to be a continuous-time state-space equation instead of a continuous-time transfer function for two reasons. First, it is easier to deal with time-delay in time domain. Second, Laguerre network naturally has a state space form in time domain. Actuator constraints are not considered in the problem formulation initially, rather they are incorporated into the scheme later in the framework of anti-windup design.

1.2 Literature Review

Historical background as well as current trends in Model Based Predictive Control (MBPC) are reviewed in this section. The concept of predictive control originated in the late seventies with the seminal papers on DMC [15] by Cutler and Ramaker and on Model Predictive Heuristic Control [61], by Richalet *et al.* The common features of predictive control are:

1. At each "present moment" t, a forecast of the process output over a long-range time horizon is made. This forecast is based on a mathematical model of the process dynamics, and on the future control scenario one *proposes* to apply from now on.

2. The control strategy is selected such that it brings the *predicted* process output back to the setpoint in the "best" way according to a specific control objective. Most often this is done by minimizing a quadratic performance index.
3. The resulting control is then applied to the process input but only at the present time. At the next sampling instant the whole procedure is repeated leading to an updated control action with corrections based on the latest measurements. This is called a receding horizon strategy.

Another school of thought in predictive control, whose objective is to design the underlying controllers in an adaptive control context, emerged almost independently at about the same time. Peterka's predictive controller [58], Ydstie's extended-horizon control [84], Mosca *et al.*'s MUSMAR [53] and the GPC [13] of Clarke *et al.* are all in this category. The continuous-time counterpart of GPC called CGPC is reported in [18]. However, the completely continuous-time design seems to limit its applicability. The structures of all the MBPC algorithms are the same but differ in details. For example, the DMC [15] uses a finite step response model and Model Predictive Heuristic Control [61] uses impulse response model while GPC [13] on the other hand uses an ARIMAX model.

Many application of MBPC are reported in the literature and several companies offer MBPC software. The survey paper by Garćia [31] *et al.* examines the relationship between several MBPC algorithms and industrial applications are also reported. A more recent paper by Richalet [62] presented two classical applications of MBPC. By the late eighties, MBPC had secured a widespread acceptance in process industry despite the lack of firm theoretical foundation, which is quite remarkable. It is acknowledged [51] that there is no useful general stability results for the original formulation of MBPC. In fact it was shown in [4] that GPC has difficulty controlling systems with nearly cancelled unstable poles and zeros. Although such kind of systems are difficult to control for any control methods, it nonetheless showed that GPC has some serious shortcomings. Bitmead *et al.* [4] suggested using the traditional infinite horizon LQG instead. The infinite horizon approach, albeit with guaranteed stability property, is less appealing in applications where some input and/or state constraints exist. A finite horizon with terminal state constraints is proposed independently by a group of researchers [14, 60, 52, 54]. The survey paper [11] by Clarke covers the most recent advances in MBPC. A bibliography of MBPC and related topics from 1965 to 1993 is also appended in that paper. A book entitled " Advances in Modêl-Based Predictive Control " [11], edited by Clarke is based on the presentations made at a conference wholly devoted to recent advances in MBPC. It is a complete collection of the latest results on MBPC. As pointed out by Clarke [11], MBPC can handle real-time state and actuator constraints in a natural way. This is an active research topic which has important practical implications. It is predicted [51] that MBPC will emerge as a versatile tool with many desirable properties and with a solid theoretical foundation.

It is worth pointing out at this point that most of the MBPC algorithms are *not* robust synthesis methods in the sense that there is no *explicit* incorporation of realistic plant uncertainty description in the problem formulation. Recent developments in the theory and application (to control) of convex optimization involving Linear Matrix Inequalities (LMI) [7] have opened a new avenue for research in MBPC. Much of the existing robust control theory can be recast in the framework of LMIs and the resulting convex optimization problem can be solved very efficiently using the recent interior-point methods. It is thus not surprising to see that results on MBPC using convex optimization (as opposed to conventional linear or quadratic programs) have begun to appear in the literature [40, 75]. This is certainly a promising research filed for MBPC.

Literature reviews on related topics such as receding horizon LQ control, Laguerre filter based modelling and control, anti-windup scheme, control of time-delay systems and continuous-time system identification based on sampled input output data will be given when these topics are introduced.

1.3 Contribution of the Thesis

The contributions of this thesis can be summarized as follows.

1. A new predictive control strategy is developed in a sampled-data framework. The resulting algorithm, SDGPC, has guaranteed stability property. Its relationship with infinite horizon LQ regulator is established clearly. SDGPC enjoys the advantage of continuous-time modeling and the flexibility of digital implementation.
2. A two-degree-of-freedom SDGPC based on optimization of two performance indices is proposed. Its servo performance and disturbance rejection performance can be tuned separately. Based on

this design, an anti-windup scheme is developed with guaranteed stability properties. The novel approach used here is to transform the nonlinear problem into a time-varying linear problem. This scheme has important practical implications as well as theoretical interests.

3. The one-degree-of-freedom SDGPC is extended to tracking system design.
4. Control of time-delay systems is treated in detail. A practically appealing Laguerre filter based adaptive SDGPC algorithm is developed.
5. An algorithm to estimate the parameters of continuous-time system based on sampled input output data is presented. Fast time-varying parameters can also be estimated under this framework. The effectiveness and the advantage of continuous-time model estimation and the SDGPC algorithm over the pure discrete-time approach are highlighted by an inverted pendulum experiment.

1.4 Outline of the Thesis

Chapter 2 presents the Sampled-Data Generalized Predictive Control algorithm SDGPC. Its relationship with infinite horizon LQ regulator and stability property are analyzed in detail. Simulation and tuning guidelines are also given by examples.

Chapter 3 extends the One-Degree-of-Freedom (ODF) SDGPC to tracking problems resulting in a Two-Degree-of-Freedom (TDF) design formulation. The TDF-SDGPC can track non-constant reference trajectories and/or disturbances with zero steady state error. When the future setpoint information is available, the TDF-SDGPC has a concise form and the tracking performance can be improved dramatically.

Chapter 4 considers control of time-delay systems. The direct approach, in which time-delay appears explicitly in the model, and Laguerre filter modeling approach are proposed. The Laguerre filter based adaptive SDGPC is particularly appealing in that its computation burden is independent on the prediction horizon.

Chapter 5 deals with another important issue in process control: actuator constraints. A SDGPC algorithm based on two performance indices is proposed. The control problem is interpreted as a nominal servo performance design plus an integrator compensation for disturbances and modeling error. This algorithm under the framework of anti-windup design effectively transforms the constrained control problem into an unconstrained time-varying control problem whose stability can be guaranteed—a pleasant result. Examples are presented to show the effectiveness of the algorithm.

Chapter 6 proposes a method to estimate the parameters of a continuous-time model based on sampled input output data. It is argued that even if the controller design is based on discrete-time model, it is always desirable to estimate the continuous-time model before discretization. An inverted pendulum is successfully controlled by SDGPC based on a continuous-time model estimated using the algorithm developed in this chapter.

Chapter 7 summarizes the thesis and gives suggestions for future research.

Chapter 2

Sampled-Data Generalized Predictive Control (SDGPC)

The poor numerical property of discrete-time models based on shift operator for fast sampling applications was shown by Middleton and Goodwin [50, pp. 44]. This is no surprise since the discrete-time model coefficients could be badly conditioned under fast sampling [50, pp. 46]. One solution is to use the δ operator. The δ operator offers superior numerical property and has great resemblance in model coefficients with its continuous-time counterpart [50, pp. 46]. Gawthrop [32] on the other hand argued that a continuous-time process is best represented by a continuous-time model and took the complete continuous-time approach, for example, in the formulation of Continuous-time Generalized Predictive Control (CGPC) [18] in which the user selected future control scenario is of a polynomial form. This approach requires approximation in digital implementation and may cause unacceptable errors for large sampling interval. The SDGPC approach given in this chapter will be based on continuous-time modeling while assuming a piecewise constant projected control scenario thus enjoying the advantages of both sides.

This chapter is organized as follows. SDGPC is formulated in section 2.1. Section 2.2 studies the stability properties of SDGPC. Section 2.3 gives interpretations for the SDGPC law in its integral form. Simulations are presented in section 2.4 to give tuning guidelines of SDGPC. Section 2.5 concludes the chapter. The work in this chapter was summarized in [46].

2.1 Formulation of SDGPC

In order to highlight the basic ideas behind SDGPC, we only consider SISO systems here. However, the extension to MIMO systems is straightforward. The system being considered is described by a state-space equation

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t) \\ y(t) &= c^T x(t) \qquad\qquad (2.1) \\ dim(x) &= n \end{aligned}$$

In order to introduce integral action in the control law, an integrator is inserted before the plant to give the augmented system

$$\begin{aligned} \dot{x}_f &= A_f x_f + B_f u_d \\ y_f &= c_f^T x_f \\ dim(x_f) &= n_f = n + 1 \end{aligned} \tag{2.2}$$

Where

$$x_d(t) = \dot{x}(t),\ \ u_d(t) = \dot{u}(t),\ \ e(t) = y(t) - w$$
$$x_f = \begin{bmatrix} x_d \\ e \end{bmatrix},\ A_f = \begin{bmatrix} A & 0 \\ c^T & 0 \end{bmatrix},\ B_f = \begin{bmatrix} B \\ 0 \end{bmatrix},\ c_f^T = [0 \ \ \cdots \ \ 0 \ \ 1] \tag{2.3}$$

And w is the constant setpoint.

We further assume that the projected future *control derivative* $u_d(t)$ is piecewise constant over the period of $T_m = \frac{T_p}{N_u}$ with values $\mathrm{u_d}(1), \mathrm{u_d}(2) \cdots \mathrm{u_d}(N_u)$ as in Fig.2.1. The benefit of assuming piecewise constant *control derivative* is that it will result in a continuous control signal.

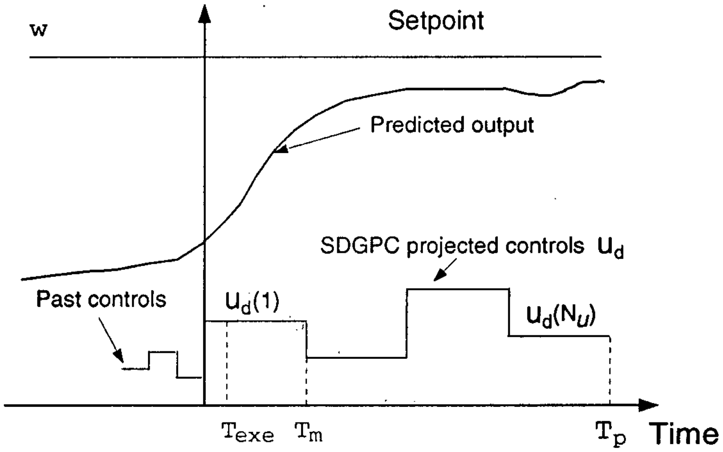

Figure 2.1: The projected control derivative

We call T_p the prediction horizon or prediction time and N_u the control order which is the allowable control maneuvers over the prediction horizon. In Fig.2.1, T_m is called the ***design sampling interval*** since the resulting SDGPC law, as will be shown in section 2.2, is equivalent to a discrete-time receding horizon control law based on (2.2) with sampling interval T_m provided that the first control $\mathrm{u_d}(1)$ is injected into the plant for a duration of T_m. However this is not necessarily so, the first control $\mathrm{u_d}(1)$ can actually be injected into the plant for a shorter time interval T_{exe} which we will call it the *execution sampling interval.* T_{exe} is the *implementation* sampling interval (in contrast to ***design*** sampling interval) and can take any value on $[0,\ T_m]$.

Similar to all other model based predictive control approaches, SDGPC is based on minimizing a performance index:

$$J(t) = \int_0^{T_p} \left[e^2(t+T) + \lambda u_d^2(t+T)\right] dT \tag{2.4}$$

Subject to :

$$x_f(t+T_p) = 0 \tag{2.5}$$

Note that the above optimization problem is a standard finite time linear quadratic regulator problem in terms of the augmented plant model (2.2).

One of the key concepts in the formulation of model based predictive control is the receding horizon strategy. However, special to SDGPC is that there are two ways to implement the receding horizon strategy. That is, after the projected control vector $[\mathrm{u_d}(1), \mathrm{u_d}(2) \cdots \mathrm{u_d}(N_u)]$ is obtained, either of the following strategies can be used:

1. The first control $\mathrm{u_d}(1)$ is applied to the plant for a time duration of T_m.
2. The first control $\mathrm{u_d}(1)$ is applied to the plant for a time duration of T_{exe} which is a fraction of the design sampling interval T_m.

The first case is equivalent to a digital control law with sampling interval T_m as will be shown in the next section. In the second case, T_{exe} can be smaller than T_m and when the execution time $T_{exe} \to 0$, it will become a continuous time control law. This approach thus has the potential to solve the numerical problem for the pure discrete-time approach, as we mentioned at the beginning of this chapter, in fast sampling applications.

With the above preparations, we are in a position to derive the SDGPC law.

The projected future control derivative in Fig.2.1 can be described mathematically as:

$$u_d(t) = H(t)\mathrm{u_d} \tag{2.6}$$

where

$$\begin{aligned} H(t) &= [H_1(t)\ \ H_2(t)\ \ \cdots H_i(t) \cdots H_{N_u}(t)] \\ \mathrm{u_d} &= [\mathrm{u_d}(1)\ \ \mathrm{u_d}(2) \cdots \mathrm{u_d}(i) \cdots \mathrm{u_d}(N_u)]^T \end{aligned} \tag{2.7}$$

$$\begin{aligned} H_i(t) &= \begin{cases} 1 & (i-1)T_m \le t < iT_m \\ 0 & otherwise \end{cases} \\ & i = 1,\ 2, \cdots N_u \\ T_m &= \frac{T_p}{N_u} \end{aligned} \tag{2.8}$$

Based on the system model (2.2) and the projected control scenario (2.6), we have the following ***T***-ahead state prediction:

$$\begin{aligned} x_d(t+T) &= e^{AT}x_d(t) + \int_0^T e^{A(T-\tau)}Bu_d(\tau)d\tau \\ &= e^{AT}x_d(t) + (\int_0^T e^{A(T-\tau)}BH(\tau)d\tau)\mathrm{u_d} \\ &= e^{AT}x_d(t) + [\int_0^T e^{A(T-\tau)}BH_1(\tau)d\tau \cdots \int_0^T e^{A(T-\tau)}BH_{N_u}(\tau)d\tau]\mathrm{u_d} \\ &= e^{AT}x_d(t) + \Gamma(T)\mathrm{u_d} \end{aligned} \tag{2.9}$$

Where

$$\Gamma(T)_{n\times N_u} = \left[\int_0^T e^{A(T-\tau)}BH_1(\tau)d\tau \cdots \int_0^T e^{A(T-\tau)}BH_{N_u}(\tau)d\tau\right]_{n\times N_u} \tag{2.10}$$

$$\Gamma(T)_{n\times N_u} = \tag{2.11}$$

$$\begin{cases} \left[\int_0^T e^{A(T-\tau)}d\tau B;\ 0;\ 0\cdots 0\right] & 0 \le T < T_m \\ \left[\int_0^{T_m} e^{A(T-\tau)}d\tau B;\ \int_{T_m}^T e^{A(T-\tau)}d\tau B;\ 0\cdots 0\right] & T_m \le T < 2T_m \\ \vdots & \vdots \\ \left[\int_0^{T_m} e^{A(T-\tau)}d\tau B; \cdots \int_{(N_u-1)T_m}^T e^{A(T-\tau)}d\tau B\right] & (N_u-1)T_m \le T < T_f \end{cases}$$

With $x_d(t+T)$, $e(t+T)$ can be obtained:

$$
\begin{aligned}
e(t+T) &= e(t) + c^T \int_0^T x_d(t+\tau)d\tau \\
&= e(t) + \int_0^T e^{A\tau} d\tau x_d(t) + c^T \Gamma_o(T)\mathrm{u}_\mathrm{d}
\end{aligned}
\tag{2.12}
$$

Where

$$
\Gamma_o(T) = \int_0^T \Gamma(\tau)d\tau \tag{2.13}
$$

Recalling the cost (2.4), we define the Hamiltonian:

$$
\begin{aligned}
H(t,\eta) &= J(t) + \eta^T x_f(t+Tp) \\
&= \int_0^{T_p} \left[e(t) + c^T A^{-1}\left(e^{AT} - I\right)x_d(t) + c^T\Gamma_o(T)\mathrm{u}_\mathrm{d}\right]^2 dT \\
&+ \int_0^{T_p} \lambda \mathrm{u}_\mathrm{d}^\mathrm{T} H^T(t)H(t)\mathrm{u}_\mathrm{d} dT + \eta^T \begin{bmatrix} e^{AT_p}x_d(t) + \Gamma(T_p)\mathrm{u}_\mathrm{d} \\ e(t) + c^T A^{-1}\left(e^{AT_p} - I\right)x_d(t) + c^T\Gamma_o(T_p)\mathrm{u}_\mathrm{d} \end{bmatrix}
\end{aligned}
\tag{2.14}
$$

Let $\frac{\partial H}{\partial \mathrm{u}_\mathrm{d}} = 0$, $\frac{\partial H}{\partial \eta} = 0$, we have the optimal solution for u_d:

$$
\mathrm{u}_\mathrm{d} = K_d x_d(t) + K_e e(t) \tag{2.15}
$$

where

$$
\begin{aligned}
K_d &= -K_1 \left\{ K_3 T_d + T_g K_2 \begin{bmatrix} e^{AT_p} \\ c^T A^{-1}\left(e^{AT_p} - I\right) \end{bmatrix} \right\} \\
K_e &= -K_1 \left\{ K_3 T_e + T_g K_2 \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \right\} \\
T_d &= \int_0^{T_p} \Gamma_o^T(T) c c^T \left(e^{AT} - I\right) A^{-1} dT,\ T_e = \int_0^{T_p} \Gamma_o^T(T) c dT \\
T_g &= \begin{bmatrix} \Gamma(T) \\ c^T \Gamma_o(T) \end{bmatrix}^T,\ K_1 = \left(\int_0^{T_p} \Gamma_o^T(T) c c^T \Gamma_o(T) dT + \lambda \int_0^{T_p} H^T H dT \right)^{-1} \\
K_2 &= \left(T_g^T K_1 T_g\right)^{-1},\ K_3 = I - T_g K_2 T_g K_1
\end{aligned}
\tag{2.16}
$$

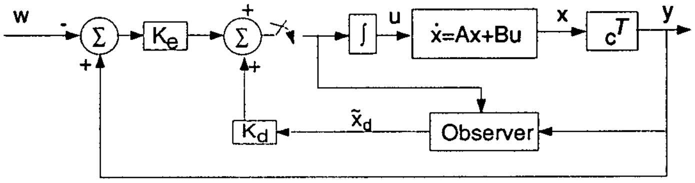

Figure 2.2: The implementation scheme

Fig.2.2 shows the SDGPC law (2.15) in a block-diagram form:

As we mentioned earlier, the control law (2.15) does not necessarily need to be implemented with the design sampling interval T_m. When the execution sampling interval T_{exe} goes to zero, we can take integration on both sides of (2.15) to obtain an *integral control law* (2.17) in terms of the state and the control signal of the original systems (2.1).

$$\mathrm{u(t)} = K_d x(t) + K_e \int^{t} e(\tau)d\tau + \eta_0 \tag{2.17}$$

The block diagram of control law (2.17) is shown in Fig. 2.3.

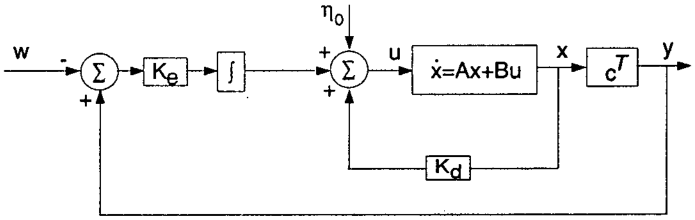

Figure 2.3: The integral control law

The constant term η_0 in (2.17) is unspecified and has no bearing on the problem in the sense that it neither affects the closed loop eigenvalue nor the asymptotic property of $e(t) \rightarrow 0$ as $t \rightarrow 0$ provided that the *integral control law* (2.17) is stabilizing. However, we can make use of the above fact and let η_0 be proportional to the constant setpoint w. The effect is that a system zero can be placed in a desired location to improve the transient response of the closed system under control law (2.17). The scheme is depicted in Fig.2.4. Details on how to select the feedforward gain K_w in Fig.2.4 will be discussed in section 2.3.

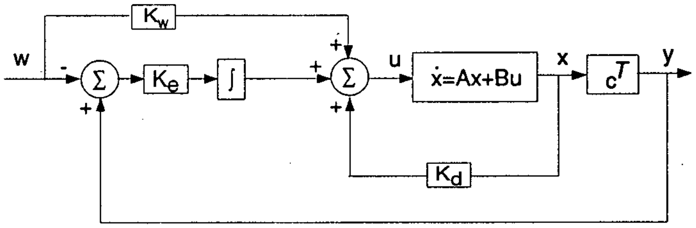

Figure 2.4: Zero placing strategy

SDGPC was developed above by minimizing the cost (2.4) subject to end point state constraints (2.5). Another approach is to include the end point state in the cost functional:

$$J(t) = \int_0^{T_p} \left[e^2(t+T) + \lambda u_d^2(t+T)\right] dT + \gamma x_f^T(t+T_p)x_f(t+T_p) \tag{2.18}$$

Substitute equations (2.6) (2.9) (2.12) into performance index (2.18), we have

$$\begin{aligned} J(t) = &\int_0^{T_p} \left[e(t) + \int_0^{T} e^{A\tau} d\tau x_d(t) + c^T\Gamma_o(T)\mathrm{u_d}\right]^2 dT + \\ &\int_0^{T_p} \lambda \mathrm{u_d}^T H^T(t)H(t)\mathrm{u_d} dT + \\ \gamma &\begin{bmatrix} e^{AT_p}x_d(t) + \Gamma(T_p)\mathrm{u_d} \\ e(t) + \int_0^{T_p} e^{A\tau} d\tau x_d(t) + c^T\Gamma_o(T_p)\mathrm{u_d} \end{bmatrix}^T \begin{bmatrix} e^{AT_p}x_d(t) + \Gamma(T_p)\mathrm{u_d} \\ e(t) + \int_0^{T_p} e^{A\tau} d\tau x_d(t) + c^T\Gamma_o(T_p)\mathrm{u_d} \end{bmatrix} \end{aligned} \tag{2.19}$$

Let $\frac{\partial J}{\partial \mathrm{u_d}} = 0$, we have the solution for $\mathrm{u_d}$:

$$\begin{aligned} \mathrm{u_d} &= -K(T_d x_d(t) + T_e e(t)) \\ K &= \left(\int_0^{T_p} \Gamma_o^T cc^T \Gamma_o dT + \lambda \int_0^{T_p} H^T H dT + \gamma\Gamma^T(T_p)\Gamma(T_p) + \gamma\Gamma_o^T(T_p)cc^T\Gamma_o(T_p)\right)^{-1} \\ T_d &= \int_0^{T_p} \Gamma_o^T cc^T A^{-1}(e^{AT} - I)dT + \gamma\Gamma^T(T_p)e^{AT_p} + \gamma\Gamma_o^T(T_p)cc^T A^{-1}(e^{AT_p} - I) \\ T_e &= \int_0^{T_p} \Gamma_o^T(T)dTc + \gamma\Gamma_o^T(T_p)c \end{aligned} \tag{2.20}$$

Where Γ, Γ_o, H are defined by equation (2.10), (2.13), (2.7) respectively. Obviously, when $\gamma \to \infty$, control laws (2.20) and (2.15) become equivalent.

The main point of this section is that by selecting the projected control derivative scenario to be piecewise constant, a predictive control law SDGPC can be designed based on continuous-time modelling without causing any difficulty for digital implementation. This is in sharp contrast to CGPC.

2.2 Stability Properties of SDGPC

Stability results for GPC with terminal states constraints (or weighting) are available both for discrete-time [11, 12, 52] and continuous-time [16]. A natural question is whether SDGPC possesses such stability properties. This question will be answered in this section. The basic idea is to show that SDGPC is equivalent to a stabilizing discrete-time receding horizon LQ control law. The important work of Bitmead *et al.* [4] is included in Appendix A for completeness. Those results are used to establish the stability property of SDGPC in section 2.2.1.

2.2.1 Stability of SDGPC with control execution time $T_{exe} = T_m$

The SDGPC stability problem is attacked by first applying a transformation to convert the SDGPC problem to a discrete-time receding horizon problem, then making use of the stability results summarized in ***Theorem*** A.10 and *Corollary A.2*. The transformation is based on the work of Levis *et al.* [43] in which the infinite horizon problem was treated.

Recall the state augmented system described by equation (2.2)

$$\begin{aligned} \dot{x}_f &= A_f x_f + B_f u_d \\ y_f &= c_f^T x_f \\ dim(x_f) &= n_f = n + 1 \end{aligned} \tag{2.21}$$

Assuming that the execution sampling interval T_{exe} under SDGPC control is the same as the design sampling interval T_m, the discrete-time equivalent of the augmented system (2.21) is then

$$\begin{aligned} x_f(i+1) &= \Phi x_f(i) + \Gamma u_d(i) \\ y_f(i) &= c_f^T x_f(i) \end{aligned} \tag{2.22}$$

With

$$\Phi = e^{A_f T_m}, \ \Gamma = \int_0^{T_m} e^{A_f \tau} B_f d\tau \tag{2.23}$$

Recall the cost functional (2.4) with $Q = c_f c_f^T$:

$$J(t) = \int_0^{T_p} \left[x_f^T(t+T)Qx_f(t+T) + u_d^T(t+T)Ru_d(t+T)\right] dT \tag{2.24}$$

Subject to : $x_f(t+T_p) = 0$

With the projected control scenario described by (2.6) as in Fig.2.1, the cost (2.24) can be expressed as the sum of N_u integrals:

$$\begin{aligned} J(t) &= \int_0^{T_p} \left[x_f^T(t+T)Qx_f(t+T) + u_d^T(t+T)Ru_d(t+T)\right] dT \\ &= \sum_{i=0}^{N_u-1} \int_{iT_m}^{(i+1)T_m} \left[x_f^T(t+T)Qx_f(t+T) + u_d^T(t+T)Ru_d(t+T)\right] dT \end{aligned} \tag{2.25}$$

Define

$$x_f(i) = x_f(t+iT_m),\ \mathrm{u_d}(i) = u_d(t+iT_m),\ i = 0, 1, \cdots N_u - 1 \tag{2.26}$$

The integrals in (2.25) can be expressed as

$$\begin{aligned} &\int_{iT_m}^{(i+1)T_m} \left[x_f^T(t+T)Qx_f(t+T) + u_d^T(t+T)Ru_d(t+T)\right] dT \\ &= \int_0^{T_m} \left[x_f^T(i+\tau)Qx_f(i+\tau) + \mathrm{u_d^T}(i+\tau)R\mathrm{u_d}(i+\tau)\right] d\tau \end{aligned} \tag{2.27}$$

The inter-sampling behavior $x_f(i+\tau)$ of system (2.22) is a function of $x(i)$ and $\mathrm{u_d}(i)$ as follows

$$x_f(i+\tau) = e^{A_f\tau}x(i) + \int_0^{\tau} e^{A_f(\tau-s)}B_f\mathrm{u_d}(i)ds \tag{2.28}$$

Substitute equation (2.28) into equation (2.27), we have

$$\begin{aligned}
&\int_{iT_m}^{(i+1)T_m} \left[x_f^T(t+T)Qx_f(t+T)+u_d^T(t+T)Ru_d(t+T)\right]dT\\
&=\int_0^{T_m} \left[x_f^T(i+\tau)Qx_f(i+\tau)+\mathrm{u}_\mathrm{d}^\mathrm{T}(i+\tau)R\mathrm{u}_\mathrm{d}(i+\tau)\right]d\tau\\
&\quad= x_f^T(i)\hat{Q}x_f(i)+2x_f^T(i)M\mathrm{u}_\mathrm{d}(i)+\mathrm{u}_\mathrm{d}^\mathrm{T}(i)\hat{R}\mathrm{u}_\mathrm{d}(i)
\end{aligned}\tag{2.29}$$

where

$$\begin{aligned}
\hat{Q}&=\int_0^{T_m} e^{A_f^T\tau}Qe^{A_f\tau}d\tau,\ M=\int_0^{T_m} e^{A_f^T\tau}Q\left[\int_0^{\tau} e^{A_ft}dt\right]B_fd\tau\\
\hat{R}&=T_mR+B_f^T\int_0^{T_m}\left[\int_0^{\tau} e^{A_f^Tt}dt\right]Q\left[\int_0^{\tau} e^{A_ft}dt\right]d\tau B_f
\end{aligned}\tag{2.30}$$

Finally the continuous-time cost (2.24) has the form

$$\begin{aligned}
J(t)&=\int_0^{T_p} \left[x_f^T(t+T)Qx_f(t+T)+u_d^T(t+T)Ru_d(t+T)\right]dT\\
&=\sum_{i=0}^{N_u-1}\left[x_f^T(i)\hat{Q}x_f(i)+2x_f^T(i)M\mathrm{u}_\mathrm{d}(i)+\mathrm{u}_\mathrm{d}^\mathrm{T}(i)\hat{R}\mathrm{u}_\mathrm{d}(i)\right]
\end{aligned}\tag{2.31}$$

Remarks:

1. These weighting matrices are time-invariant as long as T_m is constant. The symmetric and positive semi-definite or positive definite properties of Q, R are preserved in $\hat{Q}$, $\hat{R}$.
2. Even if the control weighting $R=0$ in the original cost functional, there always is a non-zero weighting term $\hat{R}$ in the equivalent discrete-time cost.

Note that there is a cross-product term in the discrete-time cost (2.25) involving $x_f(i)$ and $\mathrm{u}_\mathrm{d}(i)$. However, by some transformation [43], the cross-product term can be removed to form a standard discrete-time cost.

Define

$$\begin{aligned}
\bar{Q}&=\hat{Q}-M\hat{R}^{-1}M^T\\
\bar{\Phi}&=\Phi-\Gamma\hat{R}^{-1}M^T\\
\mathrm{v}(i)&=\hat{R}^{-1}M^Tx_f(i)+\mathrm{u}_\mathrm{d}(i)
\end{aligned}\tag{2.32}$$

By substituting equation (2.32) into system equation (2.22) and the associated cost (2.25), we obtain

$$\begin{aligned} x_f(i+1) &= \bar{\Phi} x_f(i) + \Gamma \mathrm{v}(i) \\ y_f(i) &= c_f^T x_f(i) \\ dim(x_f) &= n_f = n+1 \end{aligned} \tag{2.33}$$

and cost functional,

$$\begin{gathered} J(t) = \sum_{i=0}^{N_u-1} \left[x_f^T(i) \bar{Q} x_f(i) + \mathrm{v}^{\mathrm{T}}(i) \hat{R} \mathrm{v}(i) \right] \\ x_f(N_u) = 0 \end{gathered} \tag{2.34}$$

For clarity, the above derivation is summarized in Table 2.1

Problem Formulation	SDGPC	Discrete-time receding horizon LQ control
System equation	$\dot{x}_f = A_f x_f + B_f u_d$ $y_f = c_f^T x_f$ $dim(x_f) = n_f = n + 1$	$x_f(i+1) = \bar{\Phi} x_f(i) + \Gamma \mathrm{v}(i)$ $y_f(i) = c_f^T x_f(i)$ $dim(x_f) = n_f = n + 1$
Performance index	$J(t) =$ $\int_0^{T_p} \left[x_f^T(t+T) Q x_f(t+T) + u_d^T(t+T) R u_d(t+T)\right] dT$	$J(t) =$ $\sum_{i=0}^{N_u - 1} \left[x_f^T(i) \bar{Q} x_f(i) + \mathrm{v}^{\mathrm{T}}(i) \hat{R} \mathrm{v}(i)\right]$
Final state constraint	$x_f(T_p) = 0$	$x_f(N_u) = 0$
Relationships	$\Phi = e^{A_f T_m},\ \Gamma = \int_0^{T_m} e^{A_f^T \tau} B_f d\tau$ $\hat{Q} = \int_0^{T_m} e^{A_f^T \tau} Q e^{A_f \tau} d\tau,\ M = \int_0^{T_m} e^{A_f^T \tau} Q \left[\int_0^{\tau} e^{A_f t} dt\right] B_f d\tau$ $\hat{R} = T_m R + B_f^T \int_0^{T_m} \left[\int_0^{\tau} e^{A_f^T t} dt\right] Q \left[\int_0^{\tau} e^{A_f t} dt\right] d\tau B_f$ $\bar{Q} = \hat{Q} - M \hat{R}^{-1} M^T$ $\bar{\Phi} = \Phi - \Gamma \hat{R}^{-1} M^T$ $\mathrm{v}(i) = \hat{R}^{-1} M^T x_f(i) + \mathrm{u_d}(i)$	

Table 2.1 Comparison of SDGPC and discrete-time receding horizon LQ control

We summarize the above results as follows:

lemma 2.1

When the execution time interval T_{exe} is equal to the design sampling interval T_m, the SDGPC problem can be transformed to a standard discrete-time receding horizon LQ control problem as summarized in Table 2.1.

From ***lemma 2.1***, it is clear that the stability problem of SDGPC boils down to finding the conditions in terms of system (2.1) under which ***Theorem A.10*** holds. We have following results to serve this purpose.

Lemma 2.2 investigates the controllability and observability of the integrator augmented system (2.2). The proof of the controllability part can be found in [59]. The proof of the observability part is straightforward as given below.

lemma **2.2** (Power *et al.* [59])

If the original system (2.1) with triple $\left(A, B, c^T\right)$ *is*

a. *both controllable and observable*

b. *there is no system zeros at the origin*

then the augmented system (2.2) with triple $\left(A_f, B_f, c_f^T\right)$ *is also controllable and observable.*

Proof: The proof for controllability of (A_f, B_f) can be found in Power and Porter [59]. The observability matrix of $\left(A, c^T\right)$ is $O_{Ac^T} = \left[\ c^T; Ac^T; A^2c^T \cdots A^{n-1}c^T\right]_{n\times n}$, with $rank(O_{Ac^T}) = n$. The observability matrix of $\left(A_f, c_f^T\right)$ is $O_{A_f c_f^T} = \begin{bmatrix} 0_{1\times n} & 1 \\ O_{Ac^T} & 0_{n\times 1} \end{bmatrix}$. Obviously, $rank\left(O_{A_f c_f^T}\right) = n+1$, and the pair $\left(A_f, c_f^T\right)$ is observable. □

Remark: Condition b is intuitively obvious. If violated, there is no way that the system output of (2.1) can be driven to a nonzero setpoint. Or in terms of the augmented system (2.2), the state $e(t)$ with nonzero initial value can not be driven to the origin.

The following theorem is due to Kalman *et al* [38].

Theorem **2.1** (Kalman *et al* [38])

Let the continuous-time system (2.2) be controllable. Then the discrete-time system (2.22) is completely controllable if:

$$I_m(\lambda_i\{A\} - \lambda_j\{A\}) \neq n\frac{2\pi}{T_m} \qquad n = \pm 1, \pm 2, \ldots. \tag{2.40}$$

whenever $R_e(\lambda_i\{A\} - \lambda_j\{A\}) = 0.$

***lemma* 2.3** (Anderson *et al.* [3, pp. 354])

Assume (Φ, Γ) *given by equation (2.23) is controllable, then* $(\bar{\Phi}, \Gamma)$ *given by (2.32) is also controllable.*

Proof: The proof is obvious. Recall $\bar{\Phi} = \Phi - \Gamma \hat{R}^{-1} M^T$, the controllability of a controllable pair (Φ, Γ) can not be changed by state feedback. □

***lemma* 2.4** (Levis *et al.* [43])

$\bar{Q} \geq 0$.

Proof: Since $Q \geq 0, R > 0$, so every integrand in (2.25)

$$\begin{aligned} I_i = & \int_{iT_m}^{(i+1)T_m} \left[x_f^T(t+T)Qx_f(t+T) + u_d^T(t+T)Ru_d(t+T)\right]dT \\ & = x_f^T(i)\hat{Q}x_f(i) + 2x_f^T(i)M\mathrm{u_d}(i) + \mathrm{u_d^T}(i)\hat{R}\mathrm{u_d}(i) \end{aligned} \tag{2.41}$$

is nonnegative for any $\mathrm{u_d}(i)$. Let $\mathrm{u_d}(i) = -\hat{R}^{-1}M^T x_f(i)$, We have from equation (2.41)

$$\begin{aligned} I_i &= x_f^T(i)\hat{Q}x_f(i) + 2x_f^T(i)M\mathrm{u_d}(i) + \mathrm{u_d^T}(i)\hat{R}\mathrm{u_d}(i) \\ &= x_f^T(i)\left(\hat{Q} - \hat{R}^{-1}M^T\right)x_f(i) \\ &= x_f^T(i)\bar{Q}x_f(i) \geq 0 \end{aligned} \tag{2.42}$$

for any $x_f(i)$. So $\bar{Q} \geq 0$. □

Lemma 2.5 establishes the observability of the pair $(\bar{\Phi}, \bar{Q})$ and the observability of the augmented system (2.2). This is a special case of the results for periodic time-varying systems given by Al-Rahmani and Franklin [2] in which multi-rate control strategy is used. A simpler proof based on the H-controllability and observability concept [6] [37] is given in the following.

***lemma* 2.5**

Assume the controllability conditions of ***Theorem*** *2.1 hold, then* $(\bar{\Phi}, \bar{Q})$ *is observable if and only if the pair* $\left(A_f, c_f^T\right)$ *of equation (2.2) is observable .*

Sufficiency: Assume $(\bar{\Phi}, \bar{Q})$ is observable but (A_f, Q) is not, then there exists an eigenvalue λ of Φ associated with a nonzero eigenvector z such that $\Phi z = \lambda z$ and $Qe^{A_f t}z = 0$ for any $t > 0$

[6]. It then follows from equation (2.30) that $\hat{Q}z = 0,\ M^T z = 0$. From equation (2.32), we have $\bar{\Phi}z = \lambda z,\ \bar{Q}z = 0$. So λ is unobservable in $(\bar{\Phi}, \bar{Q})$ [37]. This contradicts the assumption.

Necessity: Assume (A_f, Q) is observable but $(\bar{\Phi}, \bar{Q})$ is not. Let λ be an unobservable eigenvalue of $\bar{\Phi}$ and $z \neq 0$ be an associated eigenvector. We have $\bar{\Phi}z = \lambda z,\ \bar{Q}z = 0$. Recall equation (2.41), let $x_f(i) = z, \mathrm{u_d}(i) = -\hat{R}^{-1}M^T z$, we have

$$\begin{aligned} I_i &= \int\limits_{iT_m}^{(i+1)T_m} \left[x_f^T(t+T)Qx_f(t+T) + u_d^T(t+T)Ru_d(t+T)\right]dT \\ &= x_f^T(i)\hat{Q}x_f(i) + 2x_f^T(i)M\mathrm{u_d}(i) + \mathrm{u_d^T}(i)\hat{R}\mathrm{u_d}(i) \\ &= z^T\bar{Q}z = 0 \end{aligned} \tag{2.43}$$

Since $Q \geq 0,\ R > 0$, equation (2.43) implies $\int\limits_0^{T_m} x_f^T(\tau)Qx_f(\tau)d\tau = \int\limits_0^{T_m} u_d^T(\tau)Ru_d(\tau)d\tau = 0$. Further, $\int\limits_0^{T_m} u_d^T(\tau)Ru_d(\tau)d\tau = z^T M\hat{R}^{-1}(T_m R)\hat{R}^{-1}M^T z = 0$. Since $\hat{R}^{-1}(T_m R)\hat{R}^{-1} > 0$, we have $M^T z = 0$. From equation (2.32), $z^T\hat{Q}z - z^T M\hat{R}^{-1}M^T z = z^T\hat{Q}z = 0$ and $\bar{\Phi}z = \left(\Phi - \Gamma\hat{R}^{-1}M^T\right)z = \Phi z = \lambda z$. From equation (2.30), $z^T\hat{Q}z = 0$ implies $Qe^{A_f t}z = 0$. But the existence of $z \neq 0$ such that $\Phi z = \lambda z,\ Qe^{A_f t}z = 0$ contradicts the observability assumption of (A_f, Q). □

Now, we are in a position to state the main stability property of SDGPC.

***Theorem* 2.2**

For systems described by equation (2.1), if

a. *The triple (A, B, c^T) is both controllable and observable.*

b. *There is no system zero at the origin.*

c. *The control execution time T_m is selected such that the condition in Theorem 2.1 is fulfilled.*

then the resulting closed loop system under SDGPC is asymptotically stable for $N_u \geq n+1$.

Proof: According to ***lemma 2.1***, SDGPC of system (2.1) is equivalent to receding horizon control of discrete-time system (2.33). Thus we need only to prove the stability of the receding horizon control problem for system (2.33) with performance index (2.34). Conditions *a.* and *b.* guarantee the controllability and observability of the integrator augmented system (2.2) according to

lemma 2.2. From condition *c.* and *Theorem 2.1*, it is apparent that the discrete-time counterpart of (2.2) given by (2.22) is also controllable and observable. Applying ***lemma 2.3-2.5***, it is obvious that $\bar{Q} \geq 0$, $\hat{R} > 0$ in (2.34), $(\bar{\Phi}, \Gamma)$ is controllable and $(\bar{\Phi}, \bar{Q})$ is observable. Apply ***Theorem A.10*** proves the theorem. □

2.2.2 Property of SDGPC with control execution time $T_{exe} < T_m$

In section 2.1, we mentioned that the *execution sampling time interval* T_{exe}, i.e. the time interval with which the plant is actually being sampled, can take any value on $[0, T_m]$. The case of $T_{exe} < T_m$ will be analyzed in this section. This strategy is very similar in spirit to the well known GPC design practice of selecting a smaller control horizon than the prediction horizon in which case the computation burden can be greatly reduced. Fig.2.5 illustrates these two closely related strategies.

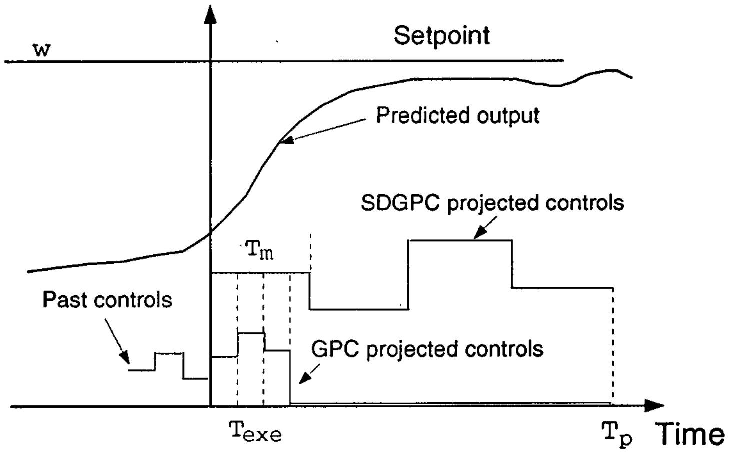

Figure 2.5: Comparison of SDGPC and GPC strategy

GPC [13] design is based on minimization of the following performance index

$$J(N) = \sum_{i=N_1}^{N_2} (y(t+i) - w(t+i))^2 + \lambda \sum_{i=0}^{N_u} [\Delta u(t+i)]^2 \tag{2.44}$$

N_1 can always be selected as zero. The prediction horizon N_2 corresponds to the prediction time T_p in SDGPC. The control weighting λ has the same meaning in SDGPC but the control horizon N_u has a quite different interpretation as is clearly illustrated in Fig.2.5. In SDGPC, N_u is the number of controls that will cover the whole prediction horizon T_p, while in GPC it is the number of controls that only cover a portion of the prediction horizon after which the control is kept constant or the increment of controls is kept zero. And in SDGPC the control execution time T_{exe} is not necessarily equal to the design sampling time T_m. It is also possible to assume the projected controls in SDGPC design have the same form as that of GPC, or any other form, say a polynomial up to certain degree. However, the advantage of choosing piecewise constant equally spaced controls over the entire prediction horizon is that in doing so the SDGPC problem can be transformed into a discrete-time receding horizon LQ problem for which powerful stability analysis methods in optimal control theory can be utilized and improved numerical property can be expected because a larger design sampling interval T_m is used.

Refer to Fig.2.5, both SDGPC and GPC use $N_u = 4$. Both SDGPC and GPC update their control every T_{exe} seconds. Both SDGPC and GPC use the same prediction horizon: $N_2 * T_{exe} = T_p$. The difference is that the design sampling interval in SDGPC is $T_m = 4 * T_{exe}$, i.e. four times as large as the execution time. Both of them have the effects of reducing computational burden and damping the control action. However, SDGPC will have superior numerical property when T_{exe} is small because SDGPC is computed based on a larger *design sampling interval* T_m while GPC is still based on T_{exe}. Another advantage of SDGPC is that although neither of them has stability claim when $N_u < N_2$ or $T_{exe} < T_m$, we know that the same SDGPC law does have guaranteed stability when $T_{exe} = T_m$. While it is very natural to choose $T_{exe} < T_m$ in SDGPC design under the framework of receding horizon strategy, it is almost unthinkable for any other controller synthesis method to design a stabilizing control law for one sampling interval but to apply it to the process with another sampling interval. It is well known that discrete-time design methods based on z-transform will encounter numerical problems when the sampling interval is small [50]. In SDGPC, a larger design sampling interval T_m can be used to improve numerical property while implementing it with a shorter sampling interval T_{exe}. Although there are no general stability results for $T_{exe} < T_m$, extensive simulation examples will be presented in next section to offer guidelines of selecting T_m and T_{exe}. As a by-product, those simulations will also shed some light on the selection of sampling interval in digital control in general.

2.3 Interpretation and Stability Property of the Integral Control Law

The *integral control law* (2.17) was obtained by integrating both sides of (2.15) under the assumption that $T_{exe} \to 0$. However, (2.17) itself can be interpreted as a solution of a well formulated predictive control problem for system (2.1). Define integral $I_e = \int^{t} (y(\tau) - w)d\tau$, where the lower limit of the integral was left blank to indicate that I_e can take any initial value, as the new state of system (2.1), the augmented system becomes:

$$\begin{aligned} \dot{x}_I &= A_f x_I + B_f u + B_\nu w \\ y &= [c^T \quad 0] x_I \\ dim(x_I) &= n+1 \end{aligned} \tag{2.45}$$

Where

$$x_I = \begin{bmatrix} x \\ I_e \end{bmatrix}, \ A_f = \begin{bmatrix} A & 0 \\ c^T & 0 \end{bmatrix}, \ B_f = \begin{bmatrix} B \\ 0 \end{bmatrix}, \ B_\nu = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tag{2.46}$$

Where w is the constant setpoint. Notice that the augmented system matrices A_f, B_f are exactly the same as of that in (2.2). The objective of the control is to let the output $y(t)$ of system (2.1) track the constant setpoint w without steady state error. Thus at equilibrium, the following relations hold:

$$\begin{aligned} \lim_{t\to\infty} y(t) &= y_\infty = w \\ \lim_{t\to\infty} u(t) &= u_\infty \\ \lim_{t\to\infty} I_e(t) &= I_\infty \\ \lim_{t\to\infty} x(t) &= x_\infty \end{aligned} \tag{2.47}$$

and

$$\begin{aligned} y_\infty = w &= c^T x_\infty \\ 0 &= A x_\infty + B u_\infty \end{aligned} \tag{2.48}$$

where $u_\infty, I_\infty, x_\infty$ are constants whose value can not be determined *a priori* based on the nominal plant parameter matrices (A, B, c^T) and the setpoint because of the unavoidable modelling errors. A sensible approach is thus to define the *shifted input*, the *shifted state* respectively as

$$\begin{aligned} u'(t) &= u(t) - u_\infty \\ x'(t) &= x(t) - x_\infty \\ I'_e(t) &= I_e(t) - I_\infty \\ y'(t) &= y(t) - w \end{aligned} \tag{2.49}$$

Solving (2.49) for u, x, I_e, y, substituting the results into (2.45), and using (2.48) it is not difficult to find that the shifted variables satisfy the equations

$$\begin{aligned} \dot{x}'_I &= A_f x'_I + B_f u' + B_\nu w \\ y' &= [c^T \quad 0]x'_I \end{aligned} \tag{2.50}$$

The *shifted* equilibrium of (2.50) is at zero as that in (2.2) and a predictive control problem can be well formulated by minimizing a quadratic performance index

$$J(t) = \int_0^{T_p} \left[[y'(t+T)]^2 + \lambda [u'(t+T)]^2 \right] dT \tag{2.51}$$

And at the end of the prediction horizon T_p, the state of (2.50) should be constraint to be zero, that is $x'_I(t+T_p) = 0$.

Although the above problem is well defined, it is still very inconvenient, to say the least, to obtain the control law due to the unknown equilibrium point $u_\infty, x_\infty, I_\infty$. A more effective formulation should thus have a model which accommodates the fact that at the equilibrium, the input, output and the state are all constant but at the same time should not explicitly have those unknown constants in the model. Taking derivative of both sides of the first equation of (2.45) will just do that. The resulting equivalent system model has the form

$$\dot{x}_f = A_f x_f + B_f u_d + B_\nu \dot{w} \tag{2.52}$$

where

$$\begin{aligned} x_d(t) &= \dot{x}(t), \;\; u_d(t) = \dot{u}(t) \\ e(t) &= y(t) - w, \; x_f = \begin{bmatrix} x_d \\ e \end{bmatrix} \end{aligned} \tag{2.53}$$

For constant setpoint as we assumed, $\dot{w} = 0$ and (2.52) is exactly the same as (2.2). The only modification needs to be made is that the observation matrix should be c_f^T in (2.3). The SDGPC problem for (2.2) and the associated performance index (2.4) can thus be interpreted as a sensible way to circumvent the unknown equilibrium difficulty encountered in the control problem defined by (2.50) and (2.51). According to *Theorem 2.2*, the control law (2.15) stabilize system (2.1). Similar results can be said about control law (2.17):

***Theorem* 2.3**

For systems described by equation (2.1) and the integral control law (2.17), if

a. *The triple $\left(A, B, c^T\right)$ is both controllable and observable.*

b. *There is no system zero at the origin.*

c. *The control execution time T_{exe} is equal to the design sampling time T_m and is selected such that the condition in Theorem 2.1 is fulfilled.*

d. *Zero-th order hold is used when applying (2.17) to system (2.1).*

then the resulting closed loop system under the integral control law (2.17) is asymptotically stable for $N_u \geq n + 1$.

Proof: When the integral control law (2.17) is applied to (2.1) with zero order hold, the resulting closed loop system matrix will be the same as that of by applying (2.15) to (2.1). This can be seen readily by comparing equations (2.2) and (2.45) considering that fact that the disturbance term $B_\nu w$ in (2.45) will not affect the stability of the closed loop system. Since system (2.1) is stable under the control of (2.15) according to *Theorem 2.2*, it will be stable as well under the control of (2.17). □

We mentioned in section 2.1 that the unspecified term in η_0 in (2.17) can be used to place a zero to improve the transient response of the closed loop system. In the following we show that there is a sound mathematical basis for doing so.

Consider system (2.52) and the cost (2.18), the T-ahead state predictor is described by (2.54) with u_d given by (2.7).

$$
\begin{aligned}
x_f(t+T) &= e^{A_f T} x_f(t) + D_\nu(T)\dot{w}(t) + \Gamma(A_f, B_f, T)\mathrm{u}_d \\
D_\nu(T) &= \int_0^{T} e^{A_f(T-\tau)} B_\nu d\tau
\end{aligned}
\tag{2.54}
$$

where $\Gamma(A_f, B_f, T)$ is given by (2.11) with A, B replaced by A_f, B_f.

Without detailed derivation, the optimal control to system (2.52) can be obtained as

$$
\mathrm{u}_d^* = K_{x_f} \begin{bmatrix} x_d(t) \\ e(t) \end{bmatrix} + K_\beta \dot{w}(t)
\tag{2.55}
$$

where

$$
\begin{aligned}
K_{x_f} &= K H_{x_f} \\
K_\beta &= K H_\beta \\
K &= \left(\int_0^{T_p} \Gamma^T(T) c_f c_f^T \Gamma(T) dT + \lambda \int_0^{T_p} H^T(T) H(T) dT + \gamma \Gamma^T(T_p)\Gamma(T_p) \right)^{-1}_{N_u \times N_u} \\
H_{x_f} &= -\left(\int_0^{T_p} \Gamma^T(T) c_f c_f^T e^{A_f T} dT + \gamma \Gamma^T(T_p) e^{A_f T_p} \right)_{N_u \times n_f} \\
H_\beta &= -\left(\int_0^{T_p} \Gamma^T(T) c_f c_f^T D_\nu(T) dT + \gamma \Gamma^T(T_p) D_\nu(T_p) \right)_{N_u \times n_\beta}
\end{aligned}
\tag{2.56}
$$

The counterpart of the optimal control sequences (2.55) with respect to system (2.45) is given by

$$
\mathrm{u}^* = K_{x_f} \begin{bmatrix} x(t) \\ \int^{t} e(\tau) d\tau \end{bmatrix} + K_\beta w(t)
\tag{2.57}
$$

The first control which is the only one being applied to the plant is

$$
\mathrm{u}^*(1) = K_x x(t) + K_e \int^{t} e(\tau) d\tau + K_\beta(1) w(t)
\tag{2.58}
$$

where K_x denotes the first n entries of the first row of the $N_u \times (n+1)$ matrix K_{x_f}, K_e is the last element of the first row of K_{x_f}.

The effect of $K_\beta(1)$ in (2.58) is to add a zero at $K_e/K_\beta(1)$ from reference $w(s)$ to output $y(s)$ [28 , p.559]. Since $K_\beta(1)$ does not affect the eigenvalues of the closed loop system matrix, meaning that it can take any value in addition to the one being computed by equation (2.56). This provides one extra degree of freedom in the design.

Example 2.3.1: In this simulation, the plant with transfer function

$$G(s) = \frac{1}{(s+1)^3} \tag{2.59}$$

is being controlled using control law (2.58) with the following design parameters

$$\begin{aligned} N_u &= 1 \\ T_p &= 6s \\ T_{exe} &= 0.1s \\ \lambda &= 10^{-4} \\ \gamma &= 100 \end{aligned} \tag{2.60}$$

The resulting feedback gains are:

$$\begin{aligned} K_x &= -[0.2839 \quad 0.8628 \quad 0.8776\,] \\ K_e &= -0.2993 \end{aligned} \tag{2.61}$$

The eigenvalues of the augmented closed-loop matrix $A_f + B_f[K_x \quad K_e]$ are at:

$$\begin{gathered} -1.0523 \pm 0.0653i \\ -0.8697 \\ -0.3096 \end{gathered} \tag{2.62}$$

Fig. 2.6 shows the control results for two different values of the feedforward term K_β, i.e. $K_\beta = 0$ and $K_\beta = \frac{K_e}{-0.3096} = 0.9668$. The latter K_β places a zero which cancels the last pole -0.3096 of the closed-loop system matrix resulting a faster response which can be seen from Fig. 2.6.

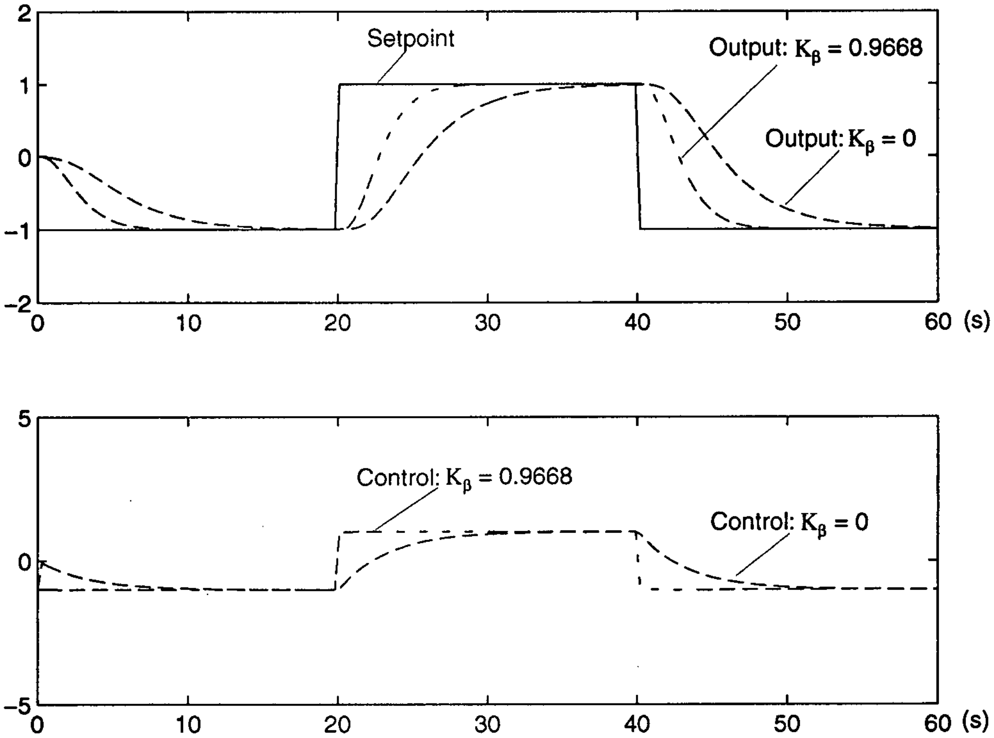

Figure 2.6: Zero placement in SDGPC

2.4 Simulations and Tuning Guidelines of SDGPC

Refer to Fig.2.7, the design parameters of SDGPC are: Prediction time T_p, design sampling interval T_m, execution sampling interval T_{exe}, and control weighting λ. The control order N_u is related to prediction time and design sampling interval by $N_u = \frac{T_p}{T_m}$. If the final states weighting is used other than final states constraint as in performance index (2.18), there is an additional design parameter γ. This is the approach used in [17] where γ served as the tuning parameter to damp the control action. However in SDGPC, we would rather fix γ to a very large value which corresponds to the states constraint case since this is crucial to guarantee stability. The task of reducing excessive control action can be accomplished by selecting $T_{exe} < T_m$, which is equivalent to putting infinite weighting on controls with sampling interval T_{exe} and only allowing control to vary every T_m time units. This will be shown later by example.

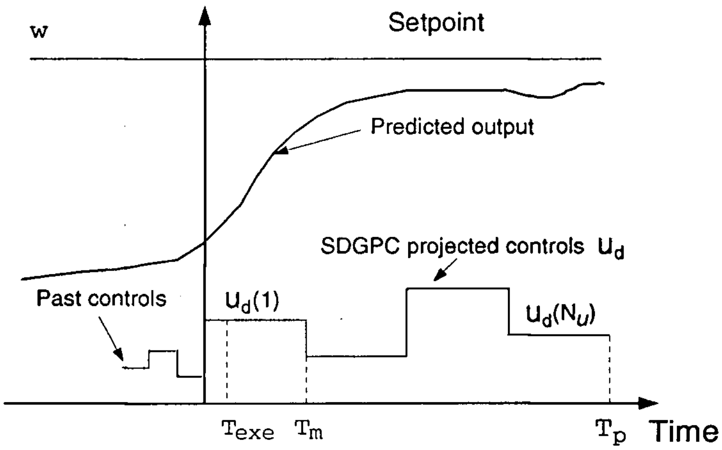

Figure 2.7: The projected control derivative

Example 1: The aim of the first example is to show the effects of the SDGPC design parameters on the control performance, and compare SDGPC with infinite horizon LQ control. The process being controlled is

$$G(s) = \frac{1}{(s+1)^3} \tag{2.63}$$

It is assumed that this process has to be controlled with a relatively fast sampling interval $T_{exe} = 0.2s$ in order to have fast disturbance rejection property. It is also assumed the states of the process is available for measurements and the derivatives of the states are computed by the state space equivalent of system model (2.63)

$$x_d(t) = \begin{bmatrix} -3 & -3 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(t) \tag{2.64}$$

Fig.2.8 shows the step response of the plant.

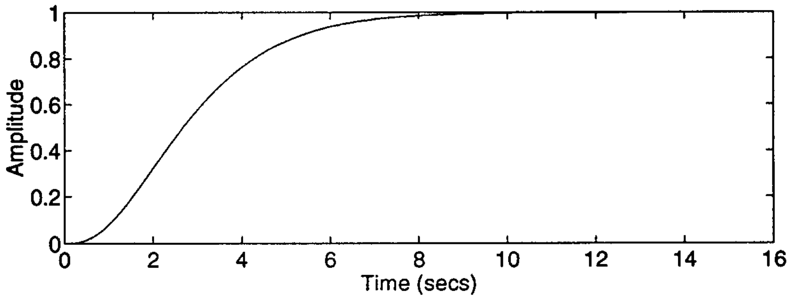

Figure 2.8: Step response of example 1

Simulation 1:

SDGPC of plant (2.63) with the following design parameters

$$\begin{aligned} N_u &= 6 \\ T_p &= 1.2s \\ T_m = T_{exe} &= 0.2s \\ \lambda &= 10^{-5}, 0.01, 0.1, 0.5, 10^{10} \end{aligned} \tag{2.65}$$

According to *Theorem 3* in section 2.2, N_u should not be smaller than 4 to ensure stability. Also from Fig.2.8, the final prediction horizon $T_p = 1.2s$ is very short for this plant.

Fig.2.9 shows the control results.

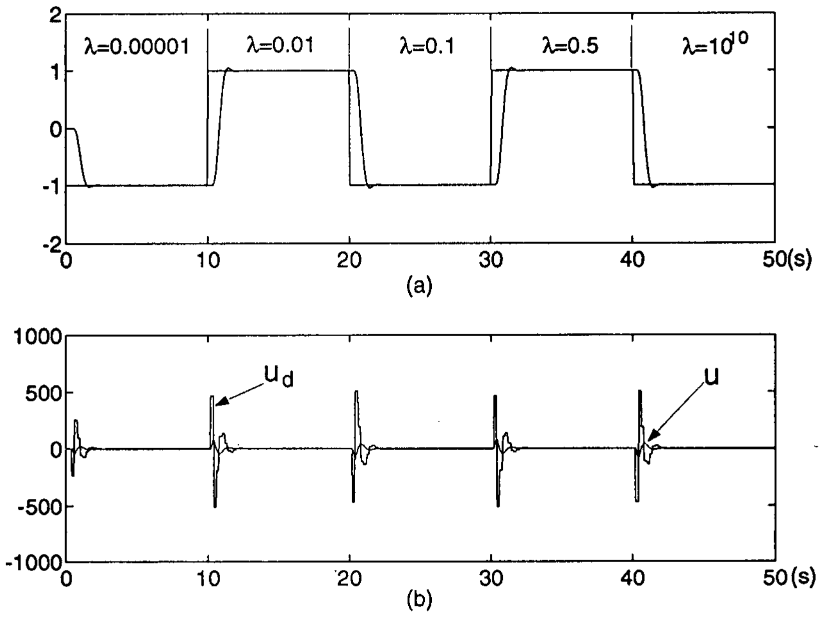

Figure 2.9: Simulation 1 of example 1

It is obvious from Fig.2.9 that this control law is unacceptable in practice because of the large magnitude of the control action. Also notice that increasing the control weighting is not effective in damping the control since the prediction horizon T_p is too short. It can be seen from Fig.2.9 that between 40 (s) and 50 (s) even a control weighting of 10^{10} can not penalize the control action. This is because when T_p is small, the end point states constraints dominate the control law calculation whereas the performance index (2.4) has little effect on the controls. Since we can not reduce the control order N_u because of the stability requirements, the only option is to increase the prediction horizon T_p. Simulation 2 shows the results.

Simulation 2:

SDGPC of plant (2.63) with

$$\begin{aligned} N_u &= 21 \\ T_p &= 4.2s \\ T_m &= T_{exe} = 0.2s \\ \lambda &= 10^{-5}, 0.01, 0.1, 0.5, 2 \end{aligned} \qquad (2.66)$$

The prediction horizon is selected to cover the significant part of the step response. See Fig.2.8. The design sampling interval and the execution sampling interval are the same as in simulation 1.

Fig.2.10 shows the results.

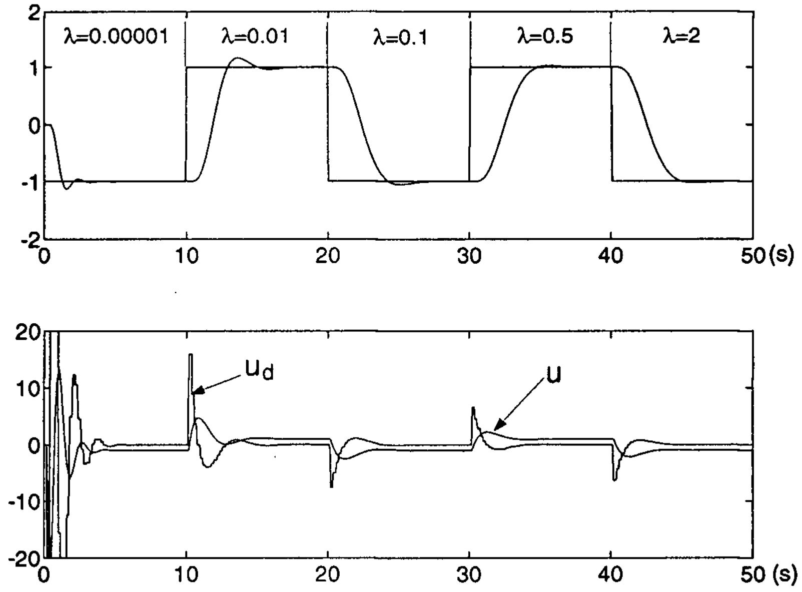

Figure 2.10: Simulation 2 of example 1

The results shown in Fig.2.10 are good except that the control law involves calculation of a 21 × 21 matrix inversion, a significant increase in computation burden compared with simulation 1.

Simulation 3:

Simulation 3 shows the SDGPC of plant (2.63) with

$$\begin{aligned} N_u &= 6 \\ T_p &= 4.2s \\ T_m &= 0.7s \\ T_{exe} &= 0.2s \\ \lambda &= 10^{-5}, 0.01, 0.1, 0.5, 2. \end{aligned} \tag{2.67}$$

The prediction horizon is the same as in simulation 2 but the control order equals the one in simulation 1. That means the design sampling interval $T_m = \frac{T_p}{N_u} = 0.7s$ and the execution sampling interval remains to be $0.2s$ as in simulation 1 and 2.

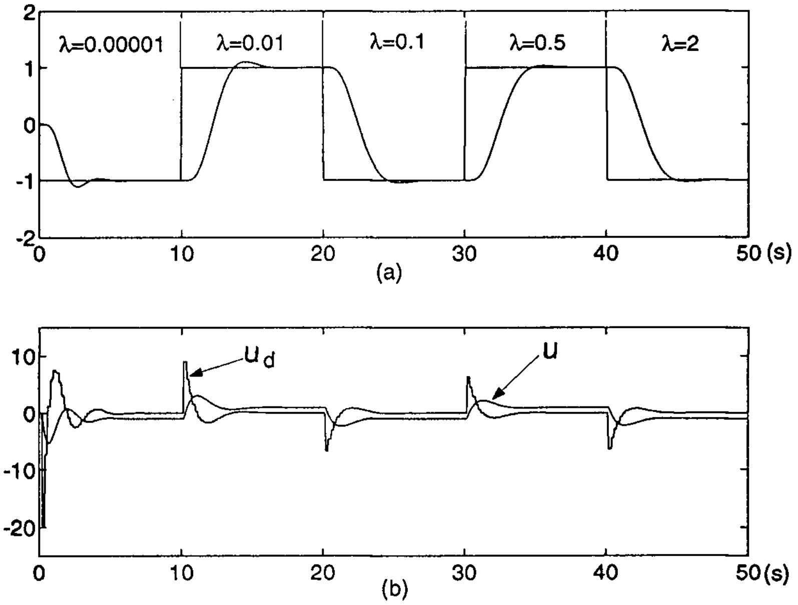

Figure 2.11: Simulation 3 of example 1

The good results in Fig.2.11 suggest that selecting $T_m > T_{exe}$ is a useful strategy to reduce computation burden and at the same time damping the control action. More simulations will be presented to support this claim in example 2.

Simulation 4:

It is interesting to compare SDGPC with infinite horizon LQR with the performance index $J = \sum_{i=0}^{\infty} \left[(y(i) - w)^2 + \lambda u_d^2(i)\right]$ in which the only tuning parameter is control weighting λ since infinite horizon is used. Plant (2.63) is discretized with sampling interval $0.2s$ as previous simulations. Control weighting varies as indicated in Fig.2.12

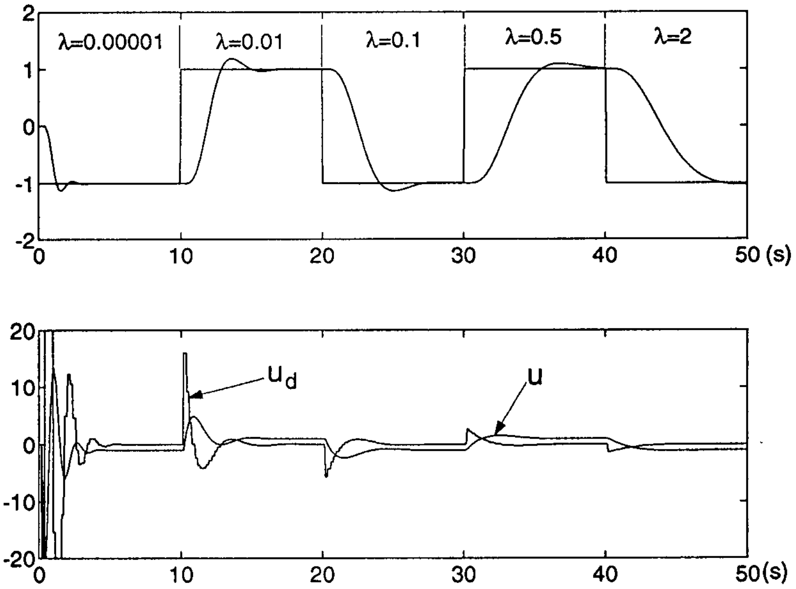

Figure 2.12: Simulation 4 of Example1: infinite horizon LQR

Compare Fig.2.12 with Fig.2.10 and Fig.2.11, it can be seen that infinite horizon LQR has visible overshoot for small control weighting λ whereas increase λ slows down the response significantly. However, this is not suggesting that SDGPC has inherent advantage over infinite horizon LQR, after all they are the same as analyzed in section 2.2.1. However, it might be easier to tune SDGPC than LQR since there are fewer design parameters in SDGPC (prediction horizon, control order etc.) than that in LQR (all entries of the weighting matrices).

Example 2: Two plants are simulated in this example. The first one is a non-minimum phase well damped open loop stable system.

$$G_1(s) = \frac{s-0.5}{(s+1)^3} \tag{2.68}$$

The second one is an open loop unstable system with imaginary poles.

$$G_2(s) = \frac{1}{(s-1)(s^2+0.4s+9)} \tag{2.69}$$

Simulation 1: Plant (2.68) is controlled by SDGPC with the following design parameters:

$$\begin{aligned} N_u &= 5 \\ T_p &= 5s \\ T_m &= 1s \\ \lambda &= 0.1 \end{aligned} \tag{2.70}$$

In the first 15 seconds, the execution sampling interval T_{exe} is equal to T_m, and after that T_{exe} is reducing every 15 seconds as illustrated in Fig.2.13

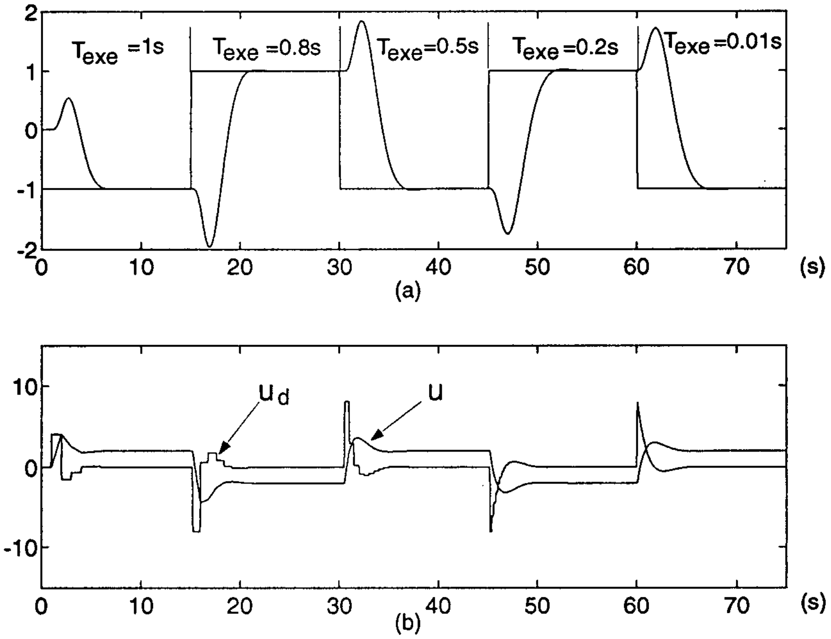

Figure 2.13: Simulation of plant (2.68)

This simulation shows that for well damped stable system (low pass plant) when fast sampling is needed, SDGPC can offer both low computation load and high implementation sampling rate by selecting $T_{exe} < T_m$.

Simulation 2: Plant (2.69) is studied. First the following group of design parameters is used.

$$\begin{aligned} N_u &= 5 \\ T_p &= 5s \\ T_m &= 1s \\ \lambda &= 0.1 \end{aligned} \tag{2.71}$$

The execution time T_{exe} varies as illustrated in Fig.2.14

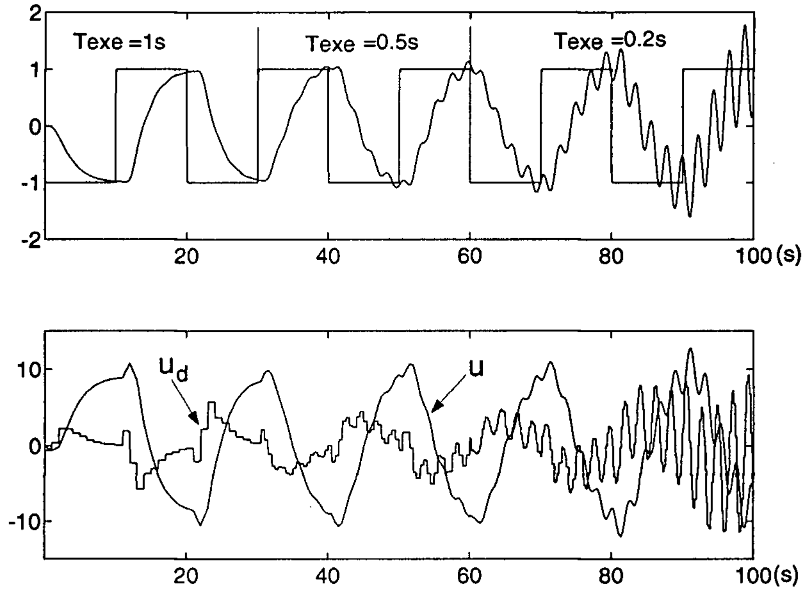

Figure 2.14: Simulation of plant (2.69)

Fig.2.14 shows that when the execution sampling interval T_{exe} is equal to the design sampling interval T_m, the performance is good. As T_{exe} decreases the performance deteriorates and the system becomes unstable when $T_{exe} = 0.2s$. Considering the plant (2.69) has an unstable pole with time constant of 1 second and has a lightly damped mode with resonance frequency of $0.4759Hz$, the design sampling interval $T_m = 1s$ is relatively large. Two things can be told by the results in Fig.2.14 for unstable and/or lightly damped systems. First, when the sampling interval T_m is relatively large, selecting $T_{exe} < T_m$ can cause performance deterioration or even instability. Second, even $T_{exe} = T_m$ is not a good choice for such a system. Since changing sampling interval can be viewed as a perturbation to the sampled-data system, the performance deterioration in Fig.2.14 means that the closed-loop system under SDGPC with sampling interval $T_m = 1s$ is sensitive to plant uncertainties.

The next simulation suggests that for systems with unstable and/or lightly damped poles the design sampling interval T_m should be at most one third of the unstable pole time constant or the sampling rate be 6 times that of the resonant frequency.

Simulation 3:

Plant (2.69) is controlled with the following design parameters.

$$\begin{gathered} N_u = 7 \\ T_p = 2.1s \\ T_m = 0.3s \\ \lambda = 0.1 \end{gathered} \tag{2.72}$$

The sampling interval T_m is reduced to one third of the unstable pole time constant. Fig.2.15 shows the results.

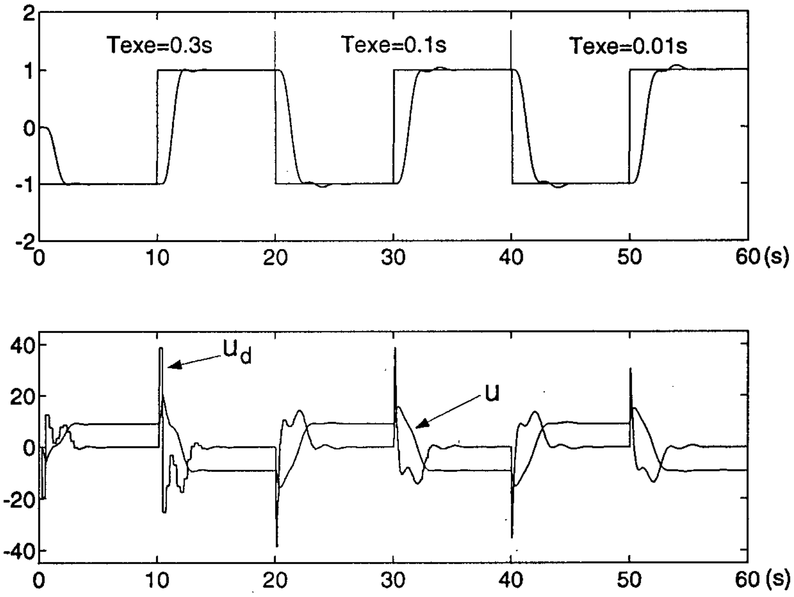

Figure 2.15: Simulation of plant (2.69)

It can be seem from Fig.2.15 that when T_m is reduced to $0.3s$ for this plant, good results are obtained regardless of the changing of the execution time.

The conclusion drawn from these two examples is that for stable well damped systems, the design parameters can be selected primarily based on performance and computation load considerations and the execution time can be selected flexibly. For unstable and/or lightly damped systems, in addition to performance and computation load considerations, there is an upper bound on the design sampling interval T_m restricted by the unstable pole time constant or the resonant frequency. There is no explicit formula available for the bound yet. But a rule of thumb is to select T_m less than one third of the unstable time constant or make $\frac{1}{T_m}$ larger than 6 times the highest resonance frequency.

Example 3:

This example shows the ability of SDGPC to control systems with nearly cancelled unstable zeros and poles. GPC will encounter difficulty controlling this kind of systems [4, pp. 102].

The plant being controlled is

$$G_2(s) = \frac{s - 0.9999}{(s-1)(2s+1)} \tag{2.73}$$

Design parameters:

$$\begin{aligned} N_u &= 6 \\ T_p &= 2.4s \\ T_m = T_{exe} &= 0.4s \\ \lambda &= 0.5 \end{aligned} \tag{2.74}$$

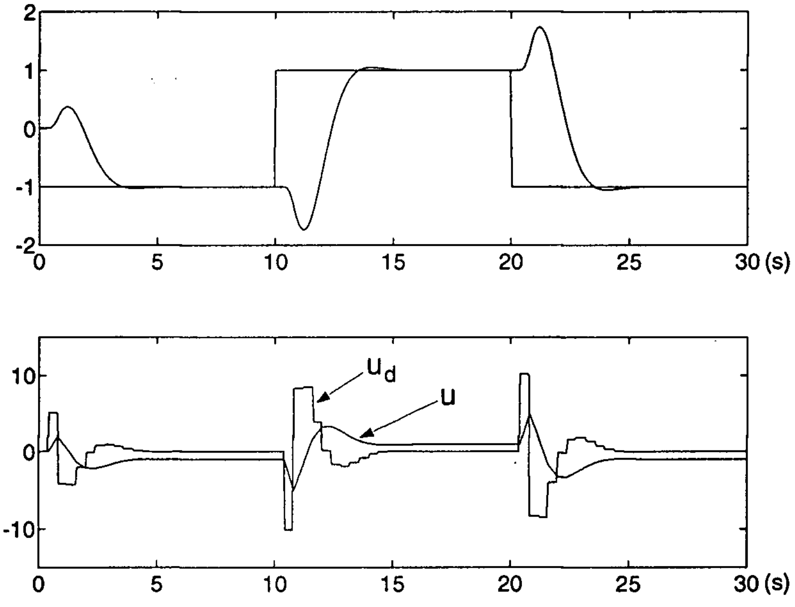

Figure 2.16: Simulation of plant (2.73)

SDGPC can control this plant without any difficulty.

2.5 Conclusion

At the beginning of this chapter, we mentioned that there are some problems with pure discrete-time approach when fast sampling is needed. A new predictive control algorithm SDGPC was thus formulated based on continuous-time modeling while assuming a piecewise constant control scenario. Better numerical property can be expected since when fast sampling is needed, the control law can actually be *designed* based on a larger sampling interval. SDGPC relates continuous-time control and discrete-time control in a natural way thus enjoys the advantage of continuous-time modeling and the flexibility of digital implementation at the same time. Under mild condition, SDGPC is shown to be equivalent to an infinite horizon LQ control law thus it has the inherent robustness property of an infinite horizon LQ regulator. However, the finite horizon formulation of SDGPC makes it convenient to handle various input and states constraints. Moreover, like other predictive control methods, the tuning of SDGPC is easy and intuition based. The design of SDGPC in this chapter is a one-degree-of-freedom design. Various extension will be made in the following chapters.

Chapter 3
SDGPC Design for Tracking Systems

In process control, the main objective is to regulate the process output at a constant level subject to various types of disturbances which is known as regulator problem. There is another type of control problem known as tracking or servo problem where it is required that the output of a system follow a desired trajectory in some optimal sense. This servo problem does occur in process industry , albeit not as often, such as the change of paper grade from a basis weight of $80g/m^2$ to $100g/m^2$ in paper production. Another important class of problem in process control which also fits into the framework of tracking control is the feed forward design problem when the disturbance information is available. The optimal tracking problem in the linear quadratic optimal control context was well formulated [41] [3]. But they were either in continuous-time or discrete-time framework. It is thus worthwhile to formulate the tracking problem in the context of sampled-data generalized predictive control. The SDGPC algorithm we developed in chapter 2 is a special case of tracking system design in which the desired trajectory is a constant setpoint. A wider class of trajectories will be considered here.

Trajectory following problems were classified in three categories in Anderson and Moore [3]. We follow the same treatment in this chapter. If the plant outputs are to follow a *class* of desired trajectories, for example, all polynomials up to a certain order, the problem is referred to as a *servo problem*; if the desired trajectory is a particular prescribed function of time, the problem is called a *tracking problem.* When the outputs of the plant are to follow the response of another plant (or model), it is referred to as the *model-following problem.* However, the differences between them are rather subtle in principle.

3.1 The Servo SDGPC Problem

Given the n-dimensional SISO linear system having state equations

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t) \\ y(t) &= c^T x(t) \end{aligned} \tag{3.75}$$

The augmented system is described by

$$\begin{aligned}\dot{x}_f &= A_f x_f + B_f u_d \\ y_f &= c_f^T x_f \\ dim(x_f) &= n_f = n+1\end{aligned} \tag{3.76}$$

Where

$$x_d(t) = \dot{x}(t), \;\; u_d(t) = \dot{u}(t)$$

$$x_f = \begin{bmatrix} x_d \\ y \end{bmatrix},\; A_f = \begin{bmatrix} A & 0 \\ c^T & 0 \end{bmatrix},\; B_f = \begin{bmatrix} B \\ 0 \end{bmatrix},\; c_f^T = [0 \;\; \cdots \;\; 0 \;\; 1] \tag{3.77}$$

Note that the system output $y(t)$ rather than the tracking error $e(t) = y(t) - r(t)$ is augmented as the system state since the setpoint $r(t)$ is no longer constrained to be constant.

Suppose the reference signal is the output of a p-dimensional linear reference model

$$\begin{aligned}\dot{w} &= Fw \\ r(t) &= R^T w\end{aligned} \tag{3.78}$$

with the pair $\left[F, R^T\right]$ completely observable.

Assume that the future projected control derivative is piecewise constant in the time interval $[t, t+T_p]$ as illustrated in Fig. 3.17, the SDGPC servo problem is to find the optimal control vector $\mathrm{u}_d = [\mathrm{u}_d(1)\; \mathrm{u}_d(2) \cdots \mathrm{u}_d(N_u)]^T$ such that the following performance index is minimized.

$$\begin{aligned}J = {} & \gamma(y(t+T_p) - r(t+T_p))^2 + \gamma x_d^T(t+T_p) x_d(t+T_p) \\ & + \int_0^{T_p} \Big[(y(t+T) - r(t+T))^2 + \lambda u_d^2(t+T)\Big] dT\end{aligned} \tag{3.79}$$

To solve this optimization problem, we need the T-ahead state prediction for both the plant (3.76) and the reference (3.78). Recall that the projected control scenario in Fig. 3.17 can be written as

$$u_d(t) = H(t)\mathrm{u}_\mathrm{d} \tag{3.80}$$

where

$$\begin{aligned}H(t) &= [H_1(t) \;\; H_2(t) \;\; \cdots H_i(t) \cdots H_{N_u}(t)] \\ \mathrm{u}_\mathrm{d} &= [\mathrm{u}_\mathrm{d}(1) \;\; \mathrm{u}_\mathrm{d}(2) \cdots \mathrm{u}_\mathrm{d}(i) \cdots \mathrm{u}_\mathrm{d}(N_u)]^T\end{aligned} \tag{3.81}$$

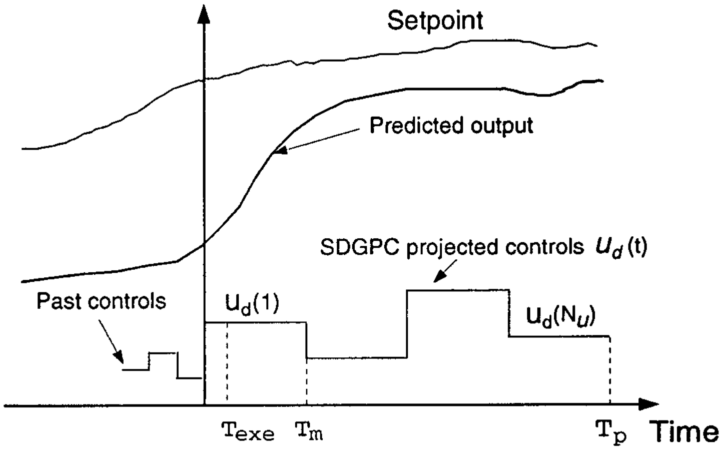

Figure 3.17: The projected control derivative

$$H_i(t) = \begin{cases} 1 & (i-1)T_m \le t < iT_m \\ 0 & otherwise \end{cases}$$
$$i = 1,\ 2, \cdots N_u \tag{3.82}$$
$$T_m = \frac{T_p}{N_u}$$

We have

$$\begin{aligned} x_f(t+T) &= e^{A_f T} x_f(t) + \int_0^T e^{A_f(T-\tau)} B_f u_d(\tau) d\tau \\ &= e^{A_f T} x_f(t) + \Gamma(A_f, B_f, T) \mathrm{u}_d \end{aligned} \tag{3.83}$$

where

$$\Gamma(A_f, B_f, T) =$$
$$\begin{cases} \left[\int_0^T e^{A_f(T-\tau)} d\tau B_f;\ 0;\ 0 \cdots 0 \right] & 0 \le T < T_m \\ \left[\int_0^{T_m} e^{A_f(T-\tau)} d\tau B_f;\ \int_{T_m}^{T} e^{A_f(T-\tau)} d\tau B_f;\ 0 \cdots 0 \right] & T_m \le T < 2T_m \\ \vdots & \vdots \\ \left[\int_0^{T_m} e^{A_f(T-\tau)} d\tau B_f; \cdots \int_{(N_u-1)T_m}^{T} e^{A_f(T-\tau)} d\tau B_f \right] & (N_u - 1)T_m \le T < T_f \end{cases} \tag{3.84}$$

And the reference state prediction is simply

$$w(t+T) = e^{FT} w(t) \tag{3.85}$$

Consequently the output predictions are as follows.

$$\begin{aligned} y(t+T) &= c_f^T x_f(t+T) \\ x_d(t+T) &= c_d^T x_f(t+T) = [I_{n\times n} 0]_{n\times n_f} x_f(t+T) \end{aligned} \tag{3.86}$$

$$r(t+T) = R^T w(t+T) \tag{3.87}$$

Substitute equations (3.86) and (3.87) into the performance index (3.79)

$$\begin{gathered} J = \\ \int_0^{T_p} \left[c_f^T e^{A_f T} x_f(t) + c_f^T \Gamma(A_f, B_f, T) \mathrm{u}_d - R^T e^{FT} w(t)\right]^2 dT \\ +\lambda \int_0^{T_p} \left[\mathrm{u}_d^T H^T(T) H(T) \mathrm{u}_d\right] dT \\ +\gamma\left[c_f^T e^{A_f T_p} x_f(t) + c_f^T \Gamma(A_f, B_f, T_p) \mathrm{u}_d - R^T e^{F T_p} w(t)\right]^2 \\ +\gamma\left[c_d^T e^{A_f T_p} x_f(t) + c_d^T \Gamma(A_f, B_f, T_p) \mathrm{u}_d\right]^T \left[c_d^T e^{A_f T_p} x_f(t) + c_d^T \Gamma(A_f, B_f, T_p) \mathrm{u}_d\right] \end{gathered} \tag{3.88}$$

Take the derivative of J with respect to u_d[1]

$$\begin{gathered} \frac{\partial J}{\partial \mathrm{u}_d} = \\ \int_0^{T_p} \left[\Gamma^T(T) c_f c_f^T \Gamma(T) \mathrm{u}_d + \Gamma^T(T) c_f \left(c_f^T e^{A_f T} x_f(t) - R^T e^{FT} w(t)\right)\right] dT \\ +\lambda \left[\int_0^{T_p} H^T(T) H(T) dT\right] \mathrm{u}_d \\ +\gamma\left[\Gamma^T(T_p) c_f c_f^T \Gamma(T_p) \mathrm{u}_d + \Gamma^T(T_p) c_f \left(c_f^T e^{A_f T_p} x_f(t) - R^T e^{F T_p} w(t)\right)\right] \\ +\gamma\left[\Gamma^T(T_p) c_d c_d^T \Gamma(T_p) \mathrm{u}_d + \Gamma^T(T_p) c_d c_d^T e^{A_f T_p} x_f(t)\right] \end{gathered} \tag{3.89}$$

The optimal SDGPC tracking control solution u_d^* is given by

$$\mathrm{u}_d^* = K_w w(t) + K_{x_f} \begin{bmatrix} x_d(t) \\ y(t) \end{bmatrix} \tag{3.90}$$

[1] A_f, B_f are dropped from $\Gamma(A_f, B_f, T)$ for clarity

where

$$
\begin{aligned}
K_w &= KH_w \\
K_{x_f} &= KH_{x_f} \\
K &= \left(\int\limits_0^{T_p} \Gamma^T(T) c_f c_f^T \Gamma(T) dT + \lambda \int\limits_0^{T_p} H^T(T) H(T) dT + \gamma \Gamma^T(T_p) \Gamma(T_p) \right)^{-1}_{N_u \times N_u} \\
H_w &= \left(\int\limits_0^{T_p} \Gamma^T(T) c_f R^T e^{FT} dT + \gamma \Gamma^T(T_p) c_f R^T e^{FT_p} \right)_{N_u \times p} \\
H_{x_f} &= - \left(\int\limits_0^{T_p} \Gamma^T(T) c_f c_f^T e^{A_f T} dT + \gamma \Gamma^T(T_p) e^{A_f T_p} \right)_{N_u \times n_f}
\end{aligned}
\tag{3.91}
$$

As shown in Fig. 3.18, the servo SDGPC law (3.90) has one feedforward term in addition to a usual feedback term as in the regulator case. This is what is known as a two-degree-of-freedom design method. Also note that equation (3.91) clearly shows that the feedback gain K_{x_f} is independent of the trajectory reference model (3.78).

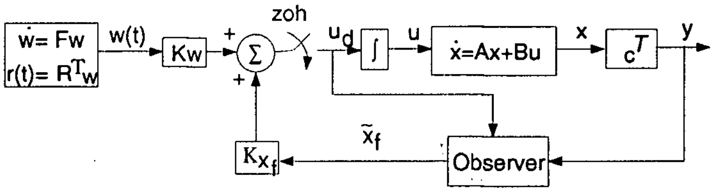

Figure 3.18: The servo SDGPC controller

So far we have assumed that the state w of the reference signal is available for measurement. In practice, however, often only an incoming signal is at hand. For this case, a state estimator may be constructed with the pair $[F, R^T]$ completely observable. Then the state estimator and the static feed forward gain K_w can be combined to give a dynamic feedforward controller as illustrated in Fig. 3.19

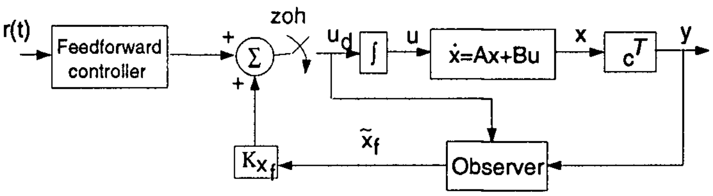

Figure 3.19: The dynamic feedforward controller implementation

When the reference model (3.78) is given by

$$F = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 1 & 0 \end{bmatrix}_{p\times p} ,R^T = [0 \quad \cdots \quad 0 \quad 1]_{1\times p} \tag{3.92}$$

the reference trajectory will be the class of signals consisting of all polynomials of degree $(p-1)$. The state $w(t)$ will be consisting of the incoming signal $r(t)$ and its derivatives up to the order $(p-1)$

$$w(t) = [\tfrac{d^{(p-1)}r(t)}{dt^{(p-1)}} \quad \cdots \quad \tfrac{dr(t)}{dt} \quad r(t)]^T \tag{3.93}$$

Clearly, the SDGPC algorithm developed in chapter 2 is a special case where $\dot{r}(t) = 0$, that is, $F = 0, R^T = 1$.

As mentioned earlier, for general F and R^T, a state estimator may be needed to construct the state of the incoming signal. However, when F and R^T are given by (3.92) with $p = 2$, a simple structure can be obtained.

According to the receding horizon strategy, only $\mathrm{u}_d^*(1)$, the first element of the optimal control u_d^*, is applied to the plant. For

$$\begin{aligned} F &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \\ R^T &= [0 \quad 1] \end{aligned} \tag{3.94}$$

, from equation (3.90), we have

$$\mathrm{u}_d^*(1) = [K_w(1,1) \quad K_w(1,2)] \begin{bmatrix} \dot{r}(t) \\ r(t) \end{bmatrix} + K_{x_f}(1,:) \begin{bmatrix} x_d(t) \\ y(t) \end{bmatrix} \tag{3.95}$$

where $K_{x_f}(1,:)$ denotes the first row of matrix K_{x_f} in equation (3.91).

A closer look at H_w and H_{x_f} in equation (3.91) reveals that the last columns of e^{FT} and e^{A_fT} are $[0 \quad T]^T$ and $[0 \quad 0 \quad \cdots \quad T]^T_{n_f\times 1}$ because the last columns of F and A_f are all zeros. Thus, the last columns of $c_fR^Te^{FT}$ in H_w is equal to the last column of matrix $c_fc_f^Te^{A_fT}$ in H_{x_f}. As an immediate consequence, $K_w(1,2)$ and $K_{x_f}(1,n_f)$ in equation 3.95 are of the same amplitude but with opposite signs. Let $K_{r_d} = K_w(1,1), K_{r,y} = K_w(1,2), K_{x_d} = K_{x_f}(1,1:n)$, we have

$$\mathrm{u}_d^*(1) = K_{r_d}\dot{r}(t) + K_{r,y}[r(t) - y(t)] + K_{x_d}x_d(t) \tag{3.96}$$

Here $K_{x_f}(1,1:n)$ denotes the first n elements of the first row of matrix K_{x_f}.

For small execution sampling time, the optimal servo SDGPC law (3.96) can be written in another form as

$$u^*(1) = K_{r_d}r(t) + K_{r,y}\int^{t}[r(\tau) - y(\tau)]d\tau + K_{x_d}x(t) \tag{3.97}$$

Compare with the optimal SDGPC solution (2.17) in chapter 2, control law (3.97) has an additional feedforward term $K_{r_d}r(t)$.

It should be noted that although the servo SDGPC solution (3.97) is derived for ramp signal $\ddot{r}(t) = 0$, it does not necessarily yield zero tracking error for a ramp even asymptotically. The reason for this is that there is only one integrator in the controller (3.97) which can only track constant reference with zero steady state error [73]. According to the *internal model principle*, there must be a model of the exogenous signal included in the control law for robust zero error tracking and disturbance rejection. As most of the discrete-time predictive control algorithms, SDGPC has the ability to track a general class of reference trajectory with zero steady state error. The spirit of this approach is state augmentation. That is, by including the equations satisfied by the external signal into the system model, a new system model in the **error space** [28] with new coordinates can be obtained and the SDGPC design procedure can then be applied. In the following, the servo SDGPC problem which incorporates double integrators in the control law is presented to show the procedure.

Consider the plant described by

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t) \\ y(t) &= c^T x(t) \\ dim(x) &= n \end{aligned} \tag{3.98}$$

The augmented system is described by

$$\begin{aligned} \dot{x}_z &= A_z x_z + B_z u_z \\ y_z &= c_z^T x_z \\ dim(x_z) &= n_z = n + 2 \end{aligned} \tag{3.99}$$

Where

$$u_z(t) = \ddot{u}(t)$$

$$x_z = \begin{bmatrix} \ddot{x} \\ \dot{y} \\ y \end{bmatrix}, \; A_z = \begin{bmatrix} A & 0_{n\times 1} & 0_{n\times 1} \\ c^T & 0 & 0 \\ 0_{1\times n} & 1 & 0 \end{bmatrix}, \; B_z = \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix}, \; c_z^T = [0 \quad \cdots \quad 0 \quad 1] \tag{3.100}$$

Assume that the reference trajectory is given by

$$\begin{aligned} \dot{w} &= Fw \\ r(t) &= R^T w \end{aligned} \tag{3.101}$$

where

$$F = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, R^T = [0 \quad 0 \quad 1], w(t) = \begin{bmatrix} \ddot{r}(t) \\ \dot{r}(t) \\ r(t) \end{bmatrix} \tag{3.102}$$

Similarly, assume that the future projected $u_z(t)$ over $[t, t+T_p]$ is piecewise constant as illustrated in Fig. 3.17, the objective is to find the optimal control vector $\mathrm{u}_z = [\mathrm{u}_z(1) \; \mathrm{u}_z(2) \cdots \mathrm{u}_z(N_u)]^T$ such that the following performance index is minimized.

$$\begin{aligned} J &= \gamma(y(t+T_p) - r(t+T_p))^2 + \gamma(\dot{y}(t+T_p) - \dot{r}(t+T_p))^2 \\ &+ \gamma \ddot{x}^T(t+T_p)\ddot{x}(t+T_p) + \int_0^{T_p} \Big[(y(t+T) - r(t+T))^2 + \lambda u_z^2(t+T)\Big] dT \end{aligned} \tag{3.103}$$

Again, we need the T-ahead state prediction for both the plant (3.99) and the reference (3.101) which are given by

$$\begin{aligned} x_z(t+T) &= e^{A_z T} x_z(t) + \int_0^T e^{A_z(T-\tau)} B_z u_z(\tau) d\tau \\ &= e^{A_z T} x_z(t) + \Gamma(A_z, B_z, T)\mathrm{u}_z \end{aligned} \tag{3.104}$$

and

$$w(t+T) = e^{FT} w(t) \tag{3.105}$$

$\Gamma(A_z, B_z, T)$ is given by equation (3.84) with A_f, B_f replaced by A_z, B_z.

The optimal solution to the performance index (3.103) is given below without detailed derivation:

$$\mathrm{u}_z^* = K_w \begin{bmatrix} \ddot{r}(t) \\ \dot{r}(t) \\ r(t) \end{bmatrix} + K_{x_z} \begin{bmatrix} \ddot{x}(t) \\ \dot{y}(t) \\ y(t) \end{bmatrix} \tag{3.106}$$

where

$$\begin{aligned} K_w &= KH_w \\ K_{x_z} &= KH_{x_z} \\ K &= \left(\int_0^{T_p} \Gamma^T(T) c_z c_z^T \Gamma(T) dT + \lambda \int_0^{T_p} H^T(T) H(T) dT + \gamma \Gamma^T(T_p) \Gamma(T_p) \right)^{-1}_{N_u \times N_u} \\ H_w &= \left(\int_0^{T_p} \Gamma^T(T) c_z R^T e^{FT} dT + \gamma \Gamma^T(T_p) \begin{bmatrix} \mathbf{0} & : \\ : & I_{2\times 2} \end{bmatrix}_{n_z \times 3} e^{FT_p} \right)_{N_u \times 3} \\ H_{x_z} &= - \left(\int_0^{T_p} \Gamma^T(T) c_z c_z^T e^{A_z T} dT + \gamma \Gamma^T(T_p) e^{A_z T_p} \right)_{N_u \times n_f} \end{aligned} \tag{3.107}$$

We are concerned about the first row of K_w and K_{x_z} since only the first element of the optimal control vector u_z^* is applied to the plant. Consider the equalities $K_w(1,2) = -K_{x_z}(1, n_z - 1), K_w(1,3) = -K_{x_z}(1, n_z)$ and let $K_r = K_w(1,1), K_{\dot{r},\dot{y}} = K_w(1,2), K_{r,y} = K_w(1,3), K_{\ddot{x}} = K_{x_z}(1, 1:n)$, the first control in (3.106) has a simplified form:

$$\begin{aligned} \mathrm{u}_z^*(1) = K_r \ddot{r}(t) + K_{\dot{r},\dot{y}}[\dot{r}(t) - \dot{y}(t)] + K_{r,y}[r(t) - y(t)] \\ + K_{\ddot{x}} \ddot{x}(t) \end{aligned} \tag{3.108}$$

Or in terms of the control input to the original plant (3.98) when the execution time goes to zero, we have by integrating both sides of equation (3.108) twice,

$$\begin{aligned} \mathrm{u}^*(1) = K_r r(t) + K_{\dot{r},\dot{y}} \int^{t} [r(\tau) - y(\tau)] d\tau + K_{r,y} \int^{t} \int^{v} [r(\tau) - y(\tau)] d\tau dv \\ + K_{\ddot{x}} x(t) \end{aligned} \tag{3.109}$$

Following is an example of servo SDGPC design to track a ramp reference signal.

Example 3.1.1 The plant being controlled has the transfer function

$$G(s) = \frac{1}{(s+1)^3} \tag{3.110}$$

Control law (3.109) is used with the design parameter

$$\begin{aligned} T_{exe} &= 0.1s \\ T_p &= 3s \\ N_u &= 6 \\ \lambda &= 0.001 \\ \gamma &= 1000 \end{aligned} \tag{3.111}$$

Fig. 3.20 shows the reference and the output, tracking error and the control input. Clearly zero steady state error is obtained.

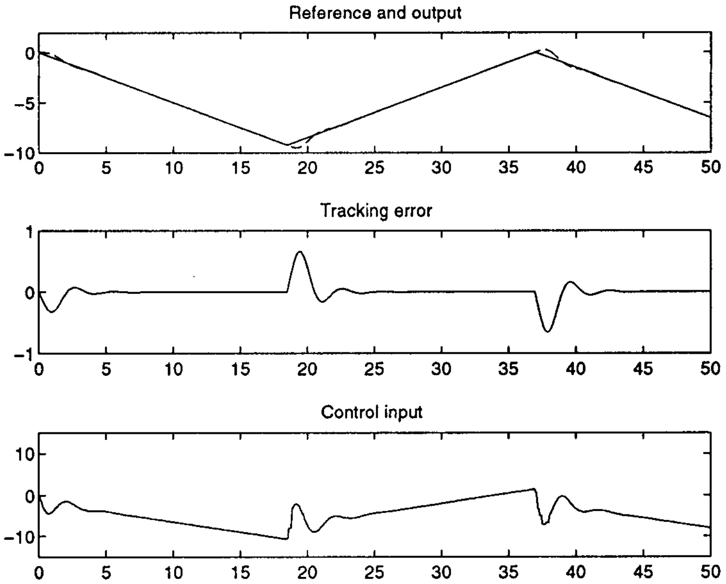

Figure 3.20: Servo SDGPC of plant (3.110)-double integrator

For comparison, the servo SDGPC law with single integrator (3.97) is designed with the same group of design parameters given in (3.111). The control results are illustrated in Fig. 3.21 in which steady state error can be clearly observed.

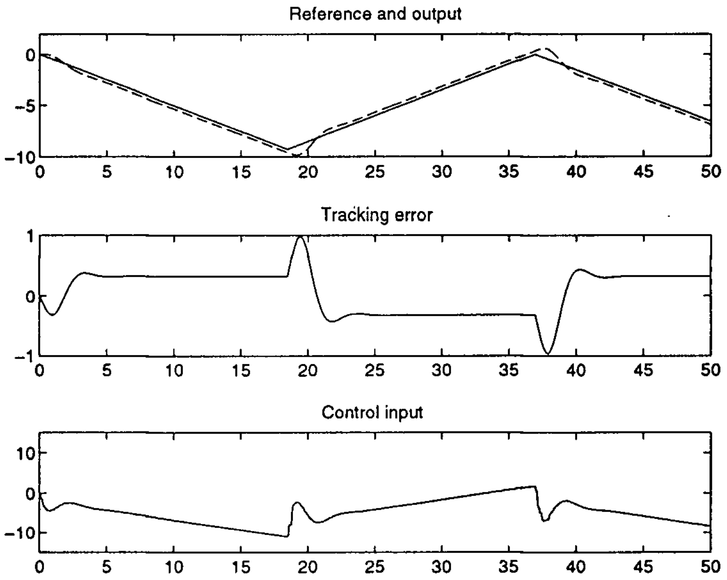

Figure 3.21: Servo SDGPC of plant (3.110)-single integrator

3.2 The Model Following SDGPC Problem

There is another kind of tracking system called the *model following problem.* It is a mild generalization of the servo problem of section 3.1. In the framework of SDGPC, the problem is to find the control vector u_d^* for the system (3.76) which minimizes the performance index (3.79), where $r(t)$ is the response of a linear system model

$$\begin{aligned} \dot{z}_1(t) &= A_1 z_1(t) + B_1 i(t) \\ r(t) &= C_1^T z_1(t) \end{aligned} \tag{3.112}$$

to command input $i(t)$, which, in turn, is the zero input response of the system

$$\begin{aligned} \dot{z}_2(t) &= A_2 z_2(t) \\ i(t) &= C_2^T z_2(t) \end{aligned} \tag{3.113}$$

as indicated in Fig. 3.22

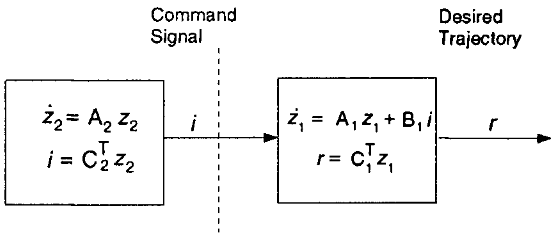

Figure 3.22: Desired trajectory for model-following problem

The two systems described by (3.112) and (3.113) can be combined into a single linear system with state space equation

$$\begin{aligned} \dot{z}(t) &= Az(t) \\ r(t) &= C^T z(t) \end{aligned} \tag{3.114}$$

where

$$z = [z_1^T \quad z_2^T],\ A = \begin{bmatrix} A_1 & B_1 C_2^T \\ \mathbf{0} & A_2 \end{bmatrix},\ C^T = [C_1^T \quad \mathbf{0}] \tag{3.115}$$

With equation (3.114), the model following problem is identical to the servo problem in section 3.1.

The following example shows the design procedure of the model following problem and the control results.

Example 3.2.1 The plant being controlled is an unstable third order process with transfer function

$$G(s) = \frac{1}{s^3 - 1} \tag{3.116}$$

The reference model has the following transfer function

$$G(s) = \frac{2}{s^2 + 4.5s + 2} \tag{3.117}$$

The step response $r(t)$ of the reference model to input c_0 is given by

$$\begin{aligned} \dot{w}(t) &= \begin{bmatrix} -4.5 & -2 \\ 1 & 0 \end{bmatrix} w(t) + \begin{bmatrix} 2 \\ 0 \end{bmatrix} c_0 \\ r(t) &= [0 \quad 1] w(t) \\ w(t) &= \begin{bmatrix} \dot{r} \\ r \end{bmatrix} \end{aligned} \tag{3.118}$$

Or in the form of equation (3.114), the above state space equation can be rewritten as

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} 0 & 0 & 0 \\ 2 & -4.5 & -2 \\ 0 & 1 & 0 \end{bmatrix} x(t) \\ r(t) &= [0 \quad 0 \quad 1] x(t) \end{aligned} \tag{3.119}$$

with $x(t) = [c_0 \quad \dot{r} \quad r]^T$.

Now the control law (3.75) can be applied. The design parameters are:

$$\begin{aligned} T_p &= 4s \\ N_u &= 20 \\ \lambda &= 0.001 \\ \gamma &= 1000 \\ T_m = T_{exe} &= 0.2s \end{aligned} \tag{3.120}$$

where T_p is the prediction horizon, N_u is the control order and λ, γ are the control weighting and final state weighting respectively. The execution sampling interval T_{exe} is set to equal to the design sampling interval T_m since the plant being controlled is unstable.

Fig. 3.23 shows the control results.

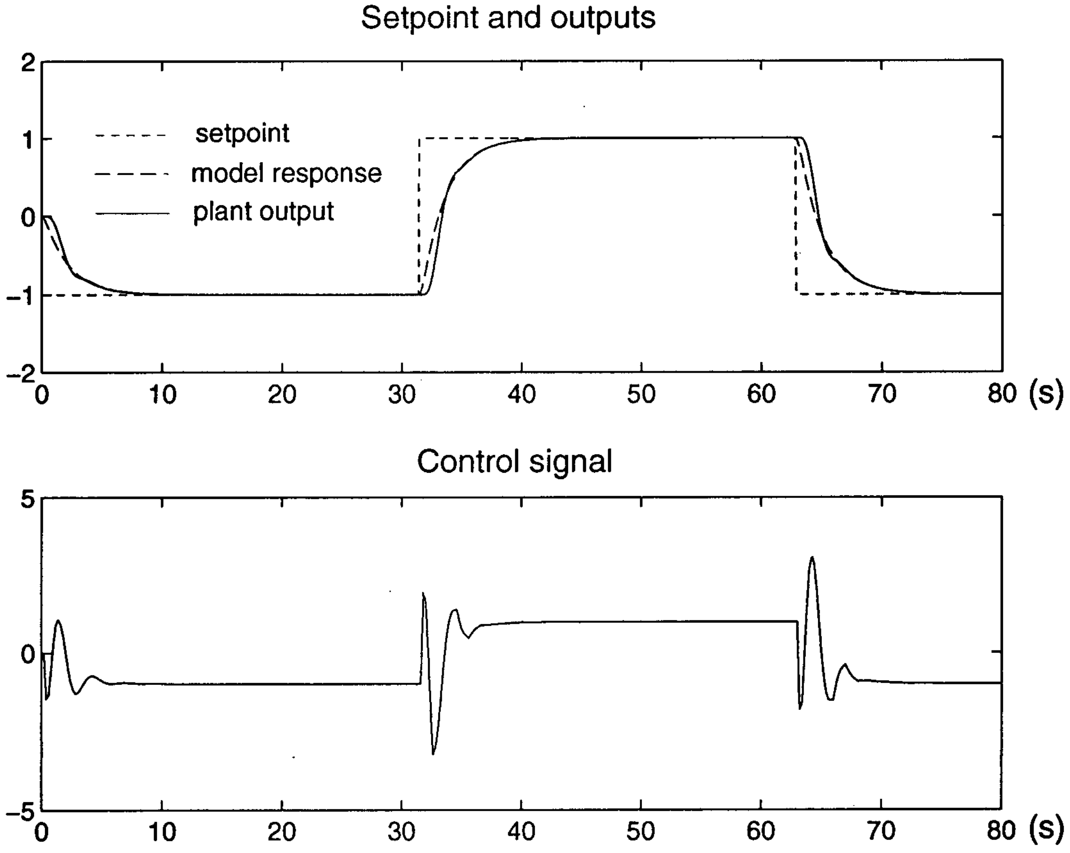

Figure 3.23: Model following control of unstable third-order system

Example 3.2.1 The second example is a third order stable plant with transfer function

$$G(s) = \frac{1}{(s+1)^3} \tag{3.121}$$

The reference model is a second order under damped plant

$$G(s) = \frac{1}{4s^2 + 2.4s + 1} \tag{3.122}$$

The design parameters are again

$$\begin{aligned} T_p &= 4s \\ N_u &= 20 \\ \lambda &= 0.001 \\ \gamma &= 1000 \\ T_m &= T_{exe} = 0.2s \end{aligned} \tag{3.123}$$

Fig. 3.24 shows the setpoint, reference model response, plant output and the control signal.

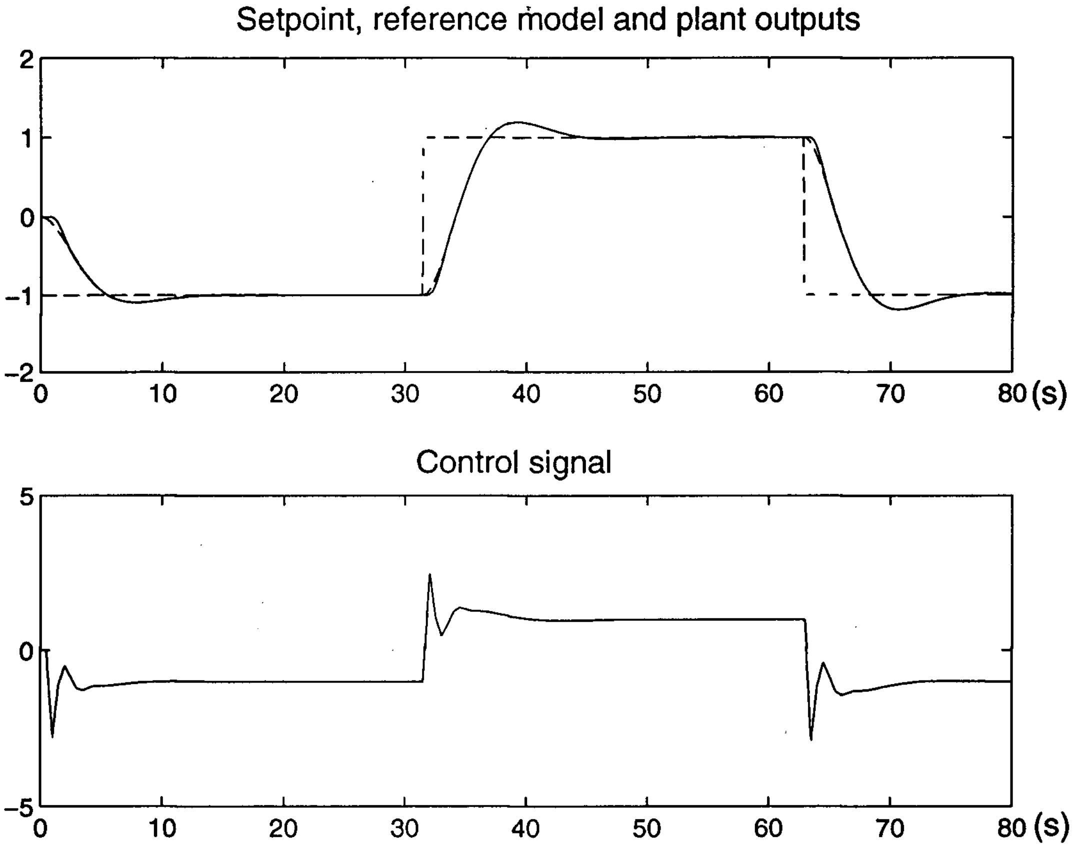

Figure 3.24: Model following control of stable third-order system

3.3 The Tracking SDGPC Problem

It is well known that when the future setpoint is available the tracking performance can be improved radically. Similarly, future values of disturbance can be utilized for better disturbance rejection. Practical examples for which future setpoints are available can be found in areas such as robot manipulator applications, high speed machining of complex shaped work pieces and vehicle lateral guidance control problems [77, 76, 57]. Predictive control is a natural candidate in these applications since it explicitly accommodates the future values of setpoint in its formulation. However, the setpoint preview capacity of predictive control has not been fully exploited before since in process control applications, where predictive control has blossomed, disturbance rejection is the major concern and the future disturbances are often unknown.

The SDGPC tracking problem is formulated as follows.

For the system (3.75) and its augmented plant (3.76), with the desired trajectory $r(t)$ available in the range $[t, t+T_p]$, the SDGPC tracking problem is to find the optimal control by minimizing the performance index (3.79).

Assuming that the projected $u_d(t)$ in $[t, t+T_p]$ is given by (3.80) as illustrated in Fig. 3.17, the performance index (3.79) can be written as

$$\begin{aligned} J = & \\ & \int_0^{T_p} \left[c_f^T e^{A_f T} x_f(t) + c_f^T \Gamma(A_f, B_f, T)\mathrm{u}_d - r(t+T)\right]^2 dT \\ & +\lambda \int_0^{T_p} \left[\mathrm{u}_d^T H^T(T) H(T) \mathrm{u}_d\right] dT \\ & +\gamma \left[c_f^T e^{A_f T_p} x_f(t) + c_f^T \Gamma(A_f, B_f, T_p)\mathrm{u}_d - r(t+T_p)\right]^2 \\ & +\gamma \left[c_d^T e^{A_f T_p} x_f(t) + c_d^T \Gamma(A_f, B_f, T_p)\mathrm{u}_d\right]^T \left[c_d^T e^{A_f T_p} x_f(t) + c_d^T \Gamma(A_f, B_f, T_p)\mathrm{u}_d\right] \end{aligned} \tag{3.124}$$

Take the derivative of J with respect to u_d, we have

$$\begin{aligned} \frac{\partial J}{\partial \mathrm{u}_d} = & \\ & \int_0^{T_p} \left[\Gamma^T(T) c_f c_f^T \Gamma(T)\mathrm{u}_d + \Gamma^T(T) c_f \left(c_f^T e^{A_f T} x_f(t) - r(t+T)\right)\right] dT \\ & +\lambda \left[\int_0^{T_p} H^T(T) H(T) dT\right] \mathrm{u}_d \\ & +\gamma \left[\Gamma^T(T_p) c_f c_f^T \Gamma(T_p)\mathrm{u}_d + \Gamma^T(T_p) c_f \left(c_f^T e^{A_f T_p} x_f(t) - r(t+T_p)\right)\right] \\ & +\gamma \left[\Gamma^T(T_p) c_d c_d^T \Gamma(T_p)\mathrm{u}_d + \Gamma^T(T_p) c_d c_d^T e^{A_f T_p} x_f(t)\right] \end{aligned} \tag{3.125}$$

The optimal SDGPC tracking control solution u_d^* is given by

$$\mathrm{u}_d^* = f_r(t) + K_{x_f} \begin{bmatrix} x_d(t) \\ y(t) \end{bmatrix} \tag{3.126}$$

where

$$
\begin{aligned}
f_r(t) &= KH_t \\
K_{x_f} &= KH_{x_f} \\
K &= \left(\int_0^{T_p} \Gamma^T(T) c_f c_f^T \Gamma(T) dT + \lambda \int_0^{T_p} H^T(T) H(T) dT + \gamma \Gamma^T(T_p) \Gamma(T_p) \right)^{-1}_{N_u \times N_u} \\
H_t &= \left(\int_0^{T_p} \Gamma^T(T) c_f r(t+T) dT + \gamma \Gamma^T(T_p) c_f r(t+T_p) \right)_{N_u \times p} \\
H_{x_f} &= -\left(\int_0^{T_p} \Gamma^T(T) c_f c_f^T e^{A_f T} dT + \gamma \Gamma^T(T_p) e^{A_f T_p} \right)_{N_u \times n_f};
\end{aligned}
\tag{3.127}
$$

With receding horizon strategy, the feedforward term $f_r(t)$ needs to be computed at every time instant. Simple numerical integration algorithm such as *Euler* approximation can be used without compromising the performance of the controller. As we mentioned at the beginning of this section, use of the future setpoint information can improve the tracking performance, sometimes significantly. The following example compares the tracking performance of two controllers one of which utilizes the future setpoint information and the other one does not.

Example 3.3.1

The plant in *Example 3.1.1* is used again with the following transfer function

$$
G(s) = \frac{1}{(s+1)^3}. \tag{3.128}
$$

First, control law (3.97) is used with the design parameter

$$
\begin{aligned}
T_{exe} &= 0.2s \\
T_p &= 6s \\
N_u &= 10 \\
\lambda &= 0.001 \\
\gamma &= 10000
\end{aligned}
\tag{3.129}
$$

Fig. 3.25 shows the setpoint and the output, the tracking error and the control input under control law (3.97).

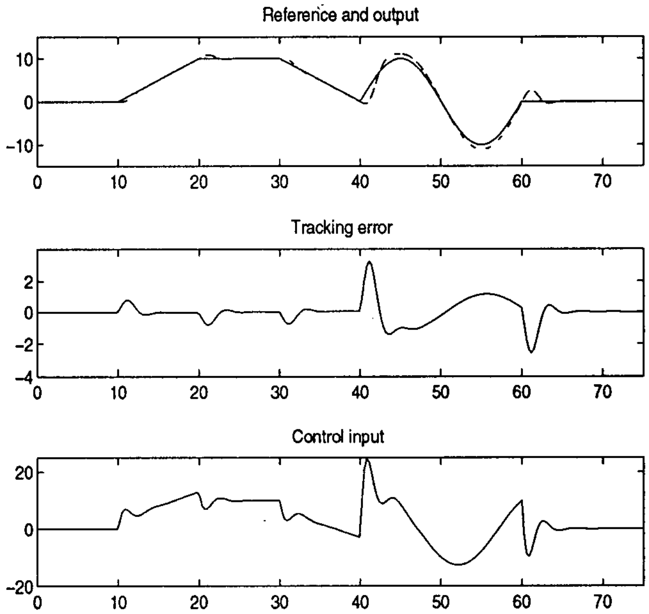

Figure 3.25: Servo SDGPC of plant (3.128)

Now the tracking control (3.126) which utilizes the future setpoint information is designed with the same design parameters given in (3.129).

Fig. 3.26 shows the results.

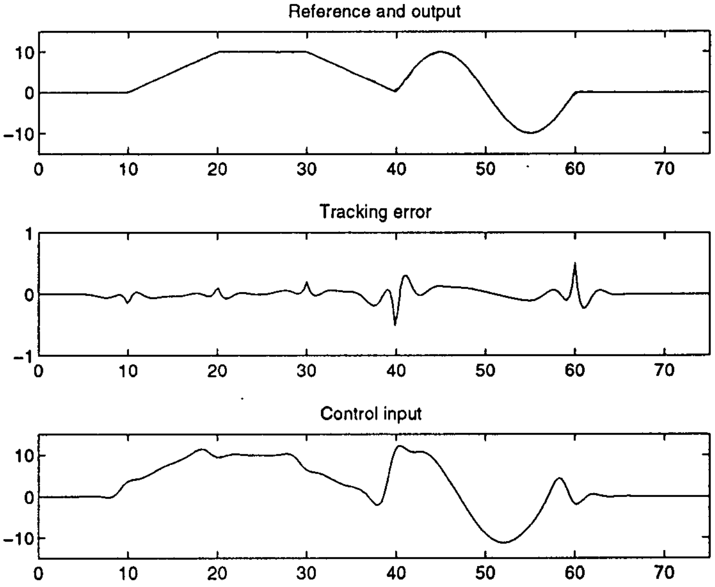

Figure 3.26: Tracking SDGPC of plant (3.128)

Compare Fig. 3.25 and Fig. 3.26, the improvements in tracking error are obvious. Also notice that the control effort in Fig. 3.26 is smoother due to the preview ability.

The improvements can be explained as follows. At current time t, knowing the future setpoint information $r(t+T)$ is equivalent to knowing the current setpoint and all its derivatives up to an arbitrarily large order. Indeed any future setpoint value $r(t+T)$ can be calculated using *Maclaurin* series expansion $r(t+T) = r(t) + \sum_{k=1}^{\infty} r^{(k)}(t)\frac{T^k}{k!}$. In control law (3.97), it was assumed that the future setpoint is a ramp. In another words, only the first derivative of the setpoint is assumed to be available. It is thus natural to expect performance improvements for complex setpoint when tracking control law (3.126) is used. However, these two control laws will not differ from each other for ramp signal. This can be confirmed by comparing Fig. 3.27 and Fig. 3.28 which show the results of plant (3.128) being controlled by (3.97) and (3.126). It can be seen that the tracking errors are the same for these two control laws at steady state while the tracking errors under control (3.126) at the transition region around time $10s$ are smaller since the setpoint here is no longer pure ramp signal.

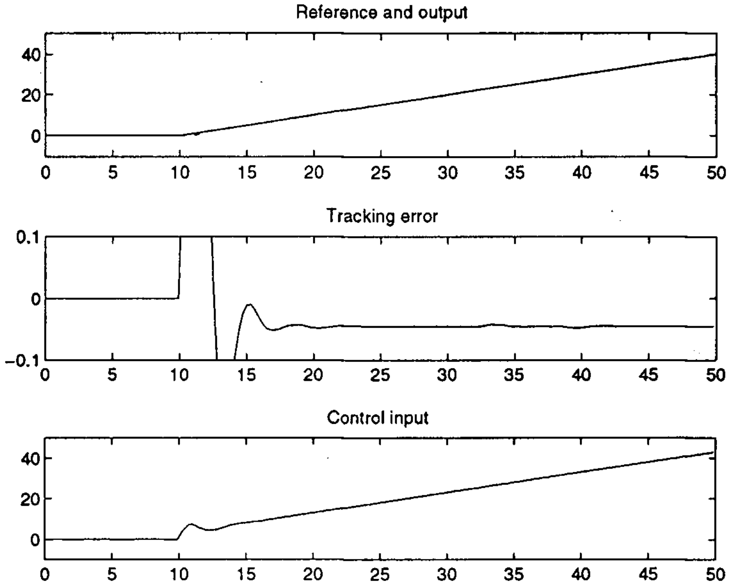

Figure 3.27: Servo SDGPC of plant (3.128)

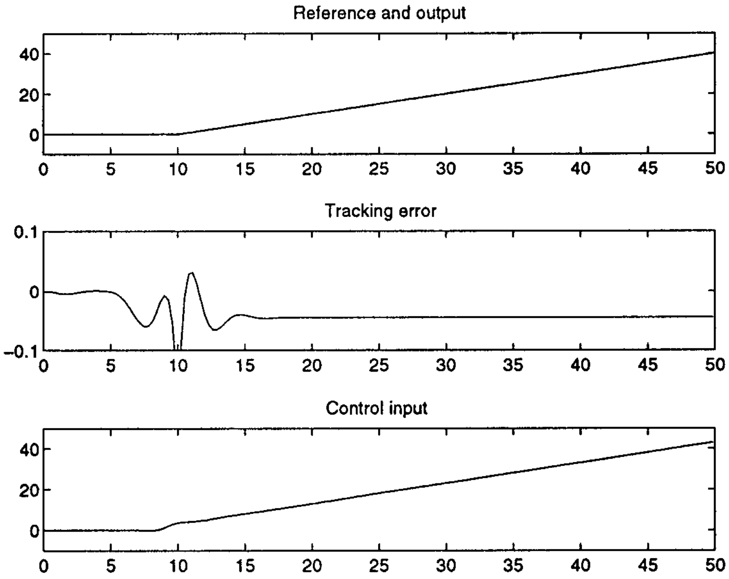

Figure 3.28: Tracking SDGPC of plant (3.128)

It should be pointed out that the tracking performance of the servo control law (3.97) can not be improved significantly by simply increasing the order of the reference model (3.94) without knowing the future setpoint information. For model order $p > 2$, the setpoint derivatives will be needed in the computation of control law (3.97). In such case a state observer of the reference model (3.94) can be constructed with desired dynamics. However, no matter how fast the dynamics of the state observer is, there is still no anticipation ability in this approach and thus the transient tracking error can not be reduced efficiently. On the other hand, the knowledge of the future setpoint can also be used in the design of control law (3.97) in which case the setpoint derivatives can be estimated using the *Maclaurin* series expansion $r(t+T) = r(t) + \sum_{k=1}^{\infty} r^{(k)}(t)\frac{T^k}{k!}$ in a least squares sense.

3.4 The Feedforward Design of SDGPC

When the disturbances can be measured, the control performance can be improved radically by utilizing this information compared with the use of feedback only. The reason is that there are inherent delays in all dynamic systems. It is always better to cancel the disturbance before it is observed at the output. Feedforward disturbance rejection also alleviates the burden of feedback disturbance rejection so that the design of the feedback loop can concentrate on robustness issues.

Here is the formulation of the feedforward design of SDGPC. Given the n-dimensional SISO linear system having state equations

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t)+B_\nu \nu(t) \\ y(t) &= c^T x(t) \end{aligned} \tag{3.130}$$

where $\nu(t)$ is a measurable disturbance satisfying state space equation

$$\begin{aligned} \dot{\beta}(t) &= W\beta(t) \\ \nu(t) &= D^T\beta(t) \end{aligned} \tag{3.131}$$

with dimension n_β.

The integrator augmented system is described by

$$\begin{aligned} \dot{x}_f &= A_f x_f + B_f u_d + B_{f_\nu}\nu_d \\ y_f &= c_f^T x_f \\ dim(x_f) &= n_f = n+1 \end{aligned} \tag{3.132}$$

Where

$$\begin{gathered} x_d(t)=\dot{x}(t),\ u_d(t)=\dot{u}(t),\ \nu_d(t)=\dot{\nu}(t) \\ x_f=\begin{bmatrix} x_d \\ y \end{bmatrix},\ A_f=\begin{bmatrix} A & \mathbf{0} \\ c^T & 0 \end{bmatrix},\ B_f=\begin{bmatrix} B \\ 0 \end{bmatrix}, B_{f_\nu}=\begin{bmatrix} B_\nu \\ 0 \end{bmatrix} \\ c_f^T=[0 \ \cdots \ 0 \ 1] \end{gathered} \tag{3.133}$$

Following the arguments in section 3.1, the T-ahead state predictor based on equation (3.132) can be written as follows:

$$\begin{aligned} x_f(t+T) &= e^{A_fT}x_f(t) + \int_0^T e^{A_f(T-\tau)}B_f u_d(\tau)d\tau + \int_0^T e^{A_f(T-\tau)}B_{f_\nu}\nu_d(\tau)d\tau \\ &= e^{A_fT}x_f(t) + \int_0^T e^{A_f(T-\tau)}B_{f_\nu}\nu_d(t+\tau)d\tau + \Gamma(A_f,B_f,T)\mathrm{u}_d \end{aligned} \tag{3.134}$$

Where $\Gamma(A_f,B_f,T)$ is given by equation (3.84), and u_d is piecewise constant as illustrated in Fig. 3.17. $\nu_d(t+\tau)$ can be obtained from state equation (3.131) as

$$\begin{aligned} \dot{\beta}(t+\tau) &= e^{W\tau}\dot{\beta}(t) \\ \nu_d(t+\tau) &= D^T e^{W\tau}\dot{\beta}(t) \end{aligned} \tag{3.135}$$

The T-ahead state predictor can be obtained by substituting equation (3.135) into (3.134):

$$\begin{aligned} x_f(t+T) &= e^{A_f T} x_f(t) + D_\nu(T)\dot{\beta}(t) + \Gamma(A_{f,} B_f, T)\mathrm{u}_d \\ D_\nu(T) &= \int_0^T e^{A_f(T-\tau)} B_{f_\nu} D^T e^{W\tau} d\tau \end{aligned} \tag{3.136}$$

With the above state predictor, the feedforward SDGPC problem is to minimize performance index (3.79) subject to the measurable disturbance $\nu(t)$.

Based on (3.136) and (3.87), the performance index (3.79) can be written as

$$\begin{aligned} J = & \\ & \int_0^{T_p} \left[c_f^T e^{A_f T} x_f(t) + c_f^T D_\nu(T)\dot{\beta}(t) + c_f^T \Gamma(T)\mathrm{u}_d - R^T e^{FT} w(t)\right]^2 dT \\ & +\lambda \int_0^{T_p} \left[\mathrm{u}_d^T H^T(T) H(T)\mathrm{u}_d\right] dT \\ & +\gamma\left[c_f^T e^{A_f T_p} x_f(t) + c_f^T D_\nu(T_p)\dot{\beta}(t) + c_f^T \Gamma(T_p)\mathrm{u}_d - R^T e^{F T_p} w(t)\right]^2 \\ & +\gamma\left[c_d^T e^{A_f T_p} x_f(t) + c_d^T D_\nu(T_p)\dot{\beta}(t) + c_d^T \Gamma(T_p)\mathrm{u}_d\right]^T \left[c_d^T e^{A_f T_p} x_f(t) + c_d^T D_\nu(T_p)\dot{\beta}(t) + c_d^T \Gamma(T_p)\mathrm{u}_d\right] \end{aligned} \tag{3.137}$$

Take the derivative of J with respect to u_d and let it be zero

$$\begin{aligned} \frac{\partial J}{\partial \mathrm{u}_d} = & \\ & \int_0^{T_p} \left[\Gamma^T(T) c_f c_f^T \Gamma(T)\mathrm{u}_d + \Gamma^T(T) c_f \left(c_f^T e^{A_f T} x_f(t) + c_f^T D_\nu(T)\dot{\beta}(t) - R^T e^{FT} w(t)\right)\right] dT \\ & +\lambda \left[\int_0^{T_p} H^T(T) H(T) dT\right] \mathrm{u}_d \\ & +\gamma\left[\Gamma^T(T_p) c_f c_f^T \Gamma(T_p)\mathrm{u}_d + \Gamma^T(T_p) c_f \left(c_f^T e^{A_f T_p} x_f(t) + c_f^T D_\nu(T_p)\dot{\beta}(t) - R^T e^{F T_p} w(t)\right)\right] \\ & +\gamma\left[\Gamma^T(T_p) c_d c_d^T \Gamma(T_p)\mathrm{u}_d + \Gamma^T(T_p) c_d c_d^T e^{A_f T_p} x_f(t) + \Gamma^T(T_p) c_d c_d^T D_\nu(T_p)\dot{\beta}(t)\right] \end{aligned} \tag{3.138}$$

The optimal SDGPC tracking control solution u_d^* is given by

$$\mathrm{u}_d^* = K_w w(t) + K_{x_f} \begin{bmatrix} x_d(t) \\ y(t) \end{bmatrix} + K_\beta \dot{\beta}(t) \tag{3.139}$$

where

$$
\begin{aligned}
K_w &= KH_w \\
K_{x_f} &= KH_{x_f} \\
K_\beta &= KH_\beta \\
K &= \left(\int_0^{T_p} \Gamma^T(T) c_f c_f^T \Gamma(T) dT + \lambda \int_0^{T_p} H^T(T) H(T) dT + \gamma \Gamma^T(T_p) \Gamma(T_p) \right)^{-1}_{N_u \times N_u} \\
H_w &= \left(\int_0^{T_p} \Gamma^T(T) c_f R^T e^{FT} dT + \gamma \Gamma^T(T_p) c_f R^T e^{FT_p} \right)_{N_u \times p} \\
H_{x_f} &= -\left(\int_0^{T_p} \Gamma^T(T) c_f c_f^T e^{A_f T} dT + \gamma \Gamma^T(T_p) e^{A_f T_p} \right)_{N_u \times n_f} \\
H_\beta &= -\left(\int_0^{T_p} \Gamma^T(T) c_f c_f^T D_\nu(T) dT + \gamma \Gamma^T(T_p) D_\nu(T_p) \right)_{N_u \times n_\beta}
\end{aligned}
\tag{3.140}
$$

Notice again that, like the setpoint feedforward term K_w, the inclusion of the disturbance feedforward termK_β does not affect the state feedback gain K_{x_f}. The control action given by (3.139) is the derivative of the control to the original plant. For small sampling interval, the direct control action can be obtained by integrating both sides of (3.139) resulting in,

$$
\mathrm{u}^*(1) = K_{x_f}(1, 1:n) x(t) + \int^t \big[K_w w(\tau) + K_{x_f}(1, n+1) y(\tau) \big] d\tau + K_\beta \beta(t) \tag{3.141}
$$

Here $K_{x_f}(1, 1:n)$ denotes the first n elements of the first row of matrix K_{x_f}.

Normally, the states $\beta(t)$ of the disturbance model (3.131) are not available. A state observer with gain L can be constructed to give the states estimates $\hat{\beta}(t)$ as follows

$$
\dot{\hat{\beta}}(t) = W\hat{\beta}(t) + L\Big(\nu(t) - D^T\hat{\beta}(t)\Big) \tag{3.142}
$$

The observer gain L is selected such that the matrix $W - LD^T$ is stable and has desired dynamics.

Following are some examples which show the effects of feedforward disturbance rejection.

Example 3.4.1

The plant being controlled is

$$G(s) = \frac{1}{(s+1)^3} \tag{3.143}$$

The disturbance is generated by a white noise passed through an integrator. However, in the design of the control law, the disturbance is assumed to be constant. That is $\dot{\nu}(t) = 0$.

The design parameters are

$$\begin{aligned} T_{exe} &= 0.2s \\ T_p &= 6s \\ N_u &= 6 \\ \lambda &= 0.001 \\ \gamma &= 1000 \end{aligned} \tag{3.144}$$

The control law takes the form of (3.141). Fig. 3.29 shows the control results. The effect of the disturbance feedforward can be seen clearly by comparing the first 50 seconds of the figure where feedforward gain K_β was set to zero and the rest of the figure where K_β is set to the value as computed.

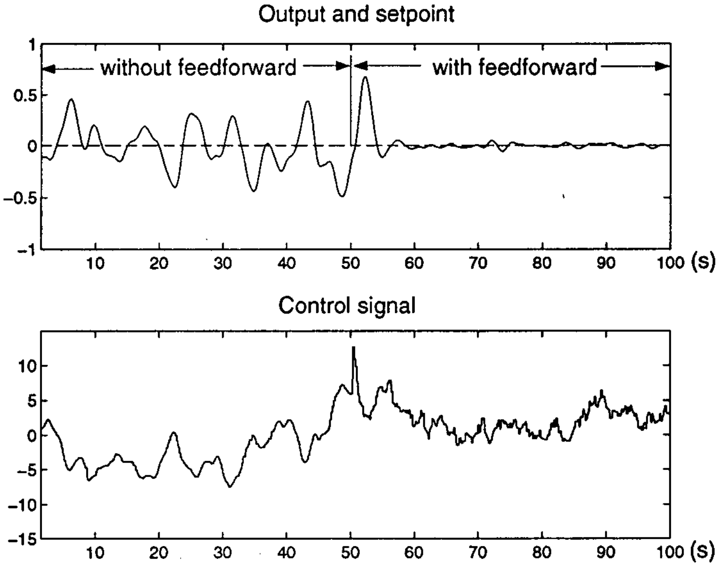

Figure 3.29: Disturbance feedforward design

This example shows how disturbance feedforward can improve the control performance dramatically even with a simple and inaccurate disturbance model. The next example shows that further performance improvements can be made with more accurate disturbance model.

Example 3.4.2

The plant being controlled is

$$G(s) = \frac{1}{s^3 + 1} \tag{3.145}$$

The disturbance is sinusoidal with known frequency. The state space model of the disturbance is:

$$\begin{aligned}\dot{\beta}(t) &= \begin{bmatrix} 0 & -4 \\ 1 & 0 \end{bmatrix} \beta(t) \\ \nu(t) &= [0 \quad 1]\beta(t)\end{aligned} \tag{3.146}$$

The observer gain $L^T = [1 \quad 4.5]$ is selected such that the eigenvalues of the observer closed loop matrix $W - LD^T$ are set to $-2, -2.5$ respectively.

The control law design parameters are

$$\begin{aligned} T_{exe} &= 0.2s \\ T_p &= 3.5s \\ N_u &= 10 \\ \lambda &= 0.001 \\ \gamma &= 1000 \end{aligned} \tag{3.147}$$

First, the correct disturbance model (3.146) was used to design the control law (3.141) and then a constant disturbance model $\dot{\nu}(t) = 0$ was used. The corresponding control results can be seen from the Fig. 3.30. As expected, the performance deteriorates in the time span between 50 seconds and 100 seconds as a wrong disturbance model is used.

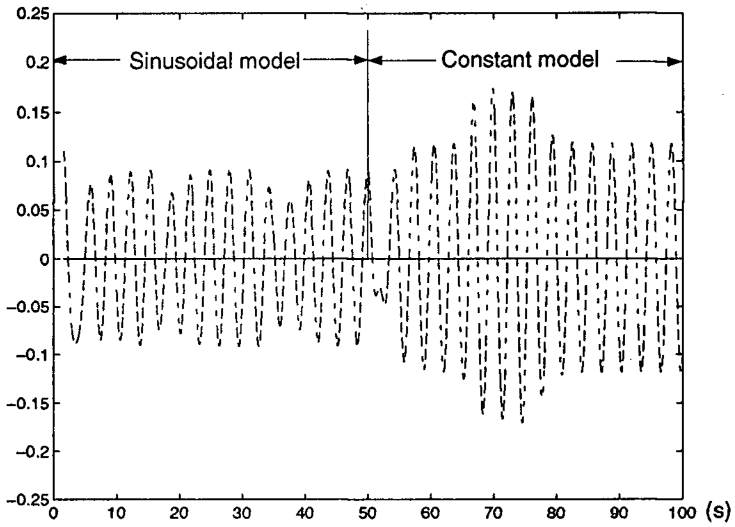

Figure 3.30: The effect of disturbance model

3.5 Conclusion

In this chapter, various tracking problems are formulated in the framework of SDGPC. Tracking control problems generally have two-degree-of-freedom design structure. However, the feedback part of the tracking problem is equivalent to the regulator problem which has one-degree-of-freedom design structure provided that the design parameters are the same. When information about the future setpoint are available, tracking performance can be radically improved. This is because knowing the future setpoint is equivalent to knowing the *exact* states information of the state equation describing the setpoint. When the disturbance is available for measurement, the disturbance rejection performance can be improved dramatically by using feedforward design.

Chapter 4

Control of Time-delay Systems and Laguerre Filter Based Adaptive SDGPC

Time-delay, or dead time, occurs frequently in industrial processes and in some cases, is time-varying. Time-delay poses one of the major challenges for the design of robust process control systems. In discrete-time, time-delay systems have the form:

$$G(q) = \frac{y(t)}{u(t)} = q^{-k} \frac{b_0 q^{-1} + b_1 q^{-2} + \cdots + b_{n-1} q^{-n}}{1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_n q^{-n}} \tag{4.148}$$

where $k = integer(T_d/\Delta)$ is the delay in samples and T_d, Δ are the delay time and sampling time respectively. For unknown time-delay, k can be either estimated directly [24] or via the extended B polynomial approach in which the leading coefficients of the B polynomial in (4.148) up to order k would be all zero. In continuous-time, time-delay can be approximated by a low order rational approximation such as Padé approximation [66]. Laguerre filter was introduced into systems theory first by Wiener in the fifties [82] and has been popular recently [87, 80, 48, 49]. In particular it can approximate time-delay systems efficiently. With the time-delay known or being modeled properly, model based predictive control strategies provide an effective way of controlling such systems. In this chapter, we give two approaches to the control of time-delay systems. The direct approach in section 4.1 is based on general state space model and assumes the time-delay is known. Emphasis will be placed on the Laguerre filter based adaptive control given in section 4.2.

4.1 The Direct Approach

The SDGPC approach to deal with time-delay systems is formulated as follows.

The system model considered is:

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t-\tau_d) \\ y(t) &= c^T x(t) \\ dim(x) &= n \end{aligned} \tag{4.149}$$

And the augmented system is given by

$$\dot{x}_f(t) = A_f x_f(t) + B_f u_d(t-\tau_d)$$
$$y_f(t) = c_f^T x_f(t) \tag{4.150}$$
$$dim(x_f) = n_f = n+1$$

Where

$$x_d(t) = \dot{x}(t), \quad u_d(t-\tau_d) = \dot{u}(t-\tau_d), \quad e(t) = y(t) - w$$
$$x_f = \begin{bmatrix} x_d \\ e \end{bmatrix}, \; A_f = \begin{bmatrix} A & 0 \\ c^T & 0 \end{bmatrix}, \; B_f = \begin{bmatrix} B \\ 0 \end{bmatrix}, \; c_f^T = [0 \quad \cdots \quad 0 \quad 1] \tag{4.151}$$

w is the constant setpoint and τ_d is the time delay in the system.

Consider the performance index

$$J(t) = \int_0^{T_p} \left[e^2(t+T) + \lambda u_d^2(t+T)\right] dT + \gamma x_f^T(t+T_p) x_f(t+T_p) \tag{4.152}$$

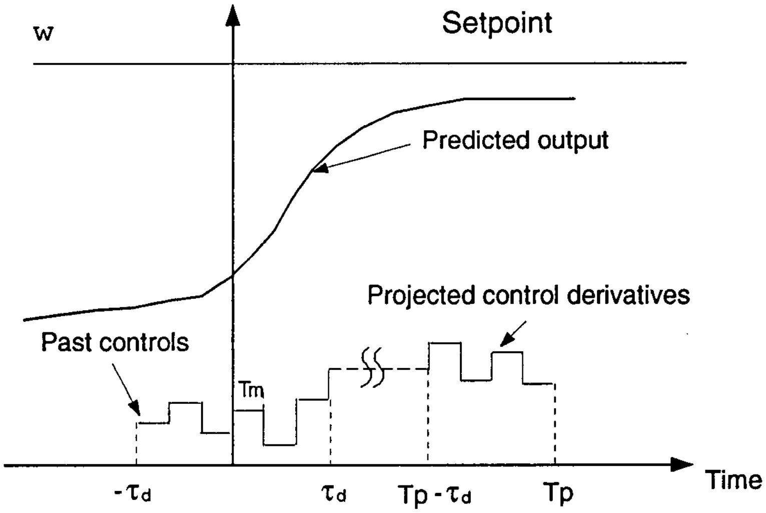

Figure 4.31: Graphical illustration of SDGPC for systems with delay

Assume the projected control signal to be piecewise constant as illustrated in Fig.4.31. For simplicity, we assume the time delay τ_d has an integral number N_d of the design sampling interval T_m. That is $N_d = \frac{T_d}{T_m}$. At present time $t = 0$, for a prediction horizon T_p, define

$$u_d(T) = H(T)\mathrm{u_d} \quad 0 \le T \le T_p - \tau_d$$
$$u_d(T) = H_{\tau_d}(T)\mathrm{u}_{\tau_d} \quad -\tau_d \le T \le 0 \tag{4.153}$$

where

$$
\begin{gathered}
H(T) = [H_1(T)\ \ H_2(T)\ \cdots H_i(T) \cdots H_{N_u}(T)] \\
H_{\tau_d}(T) = \left[H_{(-N_d)}(T)\ \ H_{(-N_d+1)}(T)\ \cdots H_{(i)}(T) \cdots H_{(-1)}(T)\right] \\
\mathrm{u_d} = [\mathrm{u_d}(1)\ \ \mathrm{u_d}(2) \cdots \mathrm{u_d}(i) \cdots \mathrm{u_d}(N_u)]^T \\
\mathrm{u}_{\tau_d} = [\mathrm{u}_{\tau_d}(-N_d)\ \ \mathrm{u}_{\tau_d}(-N_d+1) \cdots \mathrm{u}_{\tau_d}(i) \cdots \mathrm{u}_{\tau_d}(-1)]^T
\end{gathered}
\tag{4.154}
$$

$$
\begin{gathered}
H_i(T) = \begin{cases} 1 & (i-1)T_m \le T < iT_m \\ 0 & otherwise \end{cases} \\
i = 1,\ 2, \cdots N_u \\
T_m = \frac{T_p - \tau}{N_u} \\
H_{(i)}(T) = \begin{cases} 1 & iT_m \le T < (i+1)T_m \\ 0 & otherwise \end{cases} \\
i = -N_d,\ -N_d + 1, \cdots - 2, -1
\end{gathered}
\tag{4.155}
$$

With the system equation (4.150) and the control scenario (4.153), the T-ahead states prediction can be obtained:

$$
x_d(t+T) = e^{AT} x_d(t) + \Gamma_{\tau_d}(T)\mathrm{u}_{\tau_d} + \Gamma_u(T)\mathrm{u_d} \tag{4.156}
$$

where

$$
\Gamma_{\tau_d}(T) = \left[\int_{-\tau_d}^{0} e^{A(T-\tau)} B H_{(-N_d)}(\tau) d\tau \cdots \int_{-\tau_d}^{0} e^{A(T-\tau)} B H_{(-1)}(\tau) d\tau \right]_{n \times N_d} \tag{4.157}
$$

$$
\begin{gathered}
\Gamma_u(T) = \left[\int_{0}^{T-\tau_d} e^{A(T-\tau_d-\tau)} B H_1(\tau) d\tau \cdots \int_{0}^{T-\tau_d} e^{A(T-\tau_d-\tau)} B H_{N_u}(\tau) d\tau \right]_{n \times N_u} \\
T > \tau_d \\
\Gamma_u(T) = [0\ \ 0\ \ 0 \cdots 0]_{n \times N_u} \qquad 0 \le T \le \tau_d
\end{gathered}
\tag{4.158}
$$

The predicted error between system output and the setpoint is:

$$
e(t+T) = e(t) + c^T \left(\int_0^T e^{A\tau} d\tau \right) x_d(t) + c^T \Gamma_{(\tau_d)o}(T)\mathrm{u}_{\tau_d} + c^T \Gamma_{(u)o}(T)\mathrm{u_d} \tag{4.159}
$$

Where

$$\begin{aligned}
\Gamma_{(\tau_d)o}(T) &= \int_0^T \Gamma_{\tau_d}(\tau)d\tau \\
\Gamma_{(u)o}(T) &= \int_0^T \Gamma_u(\tau)d\tau
\end{aligned} \tag{4.160}$$

Substitute equations (4.156) (4.159) into cost function (4.152), the optimal control vector can be obtained as:

$$\begin{gathered}
\mathrm{u_d} = -K(T_d x_d(t) + T_e e(t) + T_{\tau_d}\mathrm{u}_{\tau_d}) \\
K = \left(\int_{\tau_d}^{T_p} \Gamma_{(u)o}^T cc^T \Gamma_{(u)o} dT + \lambda T_m I_{N_u\times N_u} + \gamma\Gamma_u^T(T_p)\Gamma_u(T_p) + \gamma\Gamma_{(u)o}^T(T_p)cc^T\Gamma_{(u)o}(T_p)\right)^{-1} \\
T_d = \int_{\tau_d}^{T_p} \Gamma_{(u)o}^T cc^T A^{-1}(e^{AT} - I)dT + \gamma\Gamma_u^T(T_p)e^{AT_p} + \gamma\Gamma_{(u)o}^T(T_p)cc^T A^{-1}(e^{AT_p} - I) \\
T_e = \int_{\tau_d}^{T_p} \Gamma_{(u)o}^T(T)dTc + \gamma\Gamma_{(u)o}^T(T_p)c \\
T_{\tau_d} = \int_{\tau_d}^{T_p} \Gamma_{(u)o}^T cc^T \Gamma_{(\tau_d)o} dT + \gamma\Gamma_u^T(T_p)\Gamma_{\tau_d}(T_p) + \gamma\Gamma_{(u)o}^T(T_p)cc^T\Gamma_{(\tau_d)o}(T_p)
\end{gathered} \tag{4.161}$$

Remark:

Systems with delay can also be treated in a LQR setting. For example, the continuous-time system model (4.150) can be first transformed into a discrete one without delay by augmenting the past controls up to time τ_d as the new system states, resulting in a system of order $N_d + n_f$ as equation (4.162). Where$N_d = \frac{\tau_d}{T_m}$, τ_d is the time-delay and T_m is the design sampling interval as illustrated in Fig.4.31.

$$\begin{bmatrix} x_f(k+1) \\ \mathrm{u}_{\tau_d}(k+1) \end{bmatrix} = \begin{bmatrix} e^{A_f T_m} & \int_0^{T_m} e^{A_f \eta} B_f d\eta & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ & & & & & 1 \\ 0 & 0 & 0 & \vdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_f(k) \\ \mathrm{u}_{\tau_d}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u_d(k) \tag{4.162}$$

$$y_f(k) = [c_f^T \quad 0] \begin{bmatrix} x_f(k) \\ \mathrm{u}_{\tau_d}(k) \end{bmatrix}$$

$$\mathrm{u}_{\tau_d}^T(k) = [u_d(k - N_d) \quad u_d(k - N_d + 1) \quad \cdots \quad u_d(k-1)]$$

Based on system equation (4.162), either finite horizon or infinite horizon discrete-time LQR solution can be found. The problem with the infinite horizon case is the singularity of the transition matrix of system (4.162) [55]. For the finite horizon case, the computation time could increase significantly due to the increase of the system order. Also notice in Fig.4.31 that although the projected controls in the time interval $[T_p - \tau_d, T_p]$ have no effect on the performance index (4.152), but due to the reverse-time iteration of the associated Riccati Difference Equation, the recursion has to go through the whole horizon whereas in the SDGPC approach , only the controls in the time interval $[0, T_p - \tau_d]$ are computed.

4.2 The Laguerre Filter Modelling Approach

The use of orthogonal series expansion, particularly Laguerre expansion has become increasingly popular in system identification and control particularly for control of systems with long and time-varying time delay. Briefly speaking, given a open loop stable system with transfer function $G(s)$, its Laguerre filter expansion is

$$G(s) = \sum_{k=1}^{\infty} c_k \frac{\sqrt{2p}}{s+p} (\frac{s-p}{s+p})^{k-1} \tag{4.163}$$

The truncated expression of equation (4.163) can be expressed in a network form as depicted in Fig.4.32.

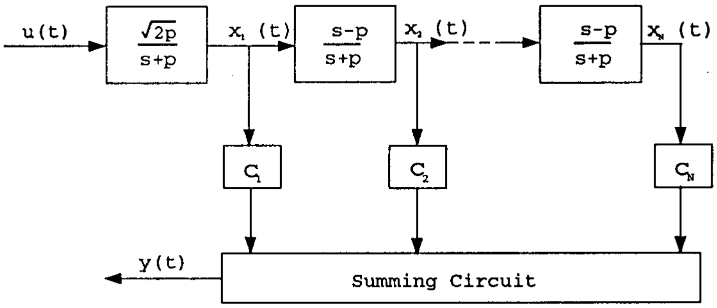

Figure 4.32: Laguerre Filter Network

Where p is the time scale selected by the user. The Laguerre network consists of a first-order low-pass filter followed by a bank of identical all-pass filters. Its input $u(t)$ is the process input. The Laguerre states defined in Fig.4.32 as $x_1(t), x_2(t), \cdots x_N(t)$ are weighted to produce the output which matches the output of the process being modeled. The set of Laguerre functions is particularly appealing because it is simple to represent, is similar to transient signals, and closely resembles Padé approximants. The continuous Laguerre functions, a complete orthonormal set in $L_2[0, \infty)$, i.e. will allow us to represent with arbitrary precision any stable system [21].

Any stable process can be expanded exactly in an infinite Laguerre series regardless of the value of the time scale p. However, when a truncated series with expansion number N is used, an immediate problem is the choice of the time scale used to ensure a fast convergence rate. Parks [56] gave a procedure to determine the optimal value of the time scale based on two measurements of the impulse response of the process being approximated. For open loop stable non-oscillatory systems with possibly long and time-varying time delays a real number p is sufficient to provide good convergence results. Fu and Dumont [29] gave an optimal time scale for discrete-time Laguerre network and proposed an optimization algorithm to search for the optimal complex time scale p when the process being modeled is highly under damped.

Since the Laguerre network has a state space representation, any state space design method can be applied to the controller design. Dumont and Zervos [22] proposed a simple one step ahead predictive controller based on discrete-time Laguerre filters. This algorithm has been commercialized and is routinely used in pulp and paper mills [21]. Dumont, Fu and Lu [23] proposed a GPC algorithm based on nonlinear Laguerre model in which the linear dynamic part is represented by a series of Laguerre filters and the nonlinear part is a memoryless nonlinear mapping. Simulation shows it has superior performance over the linear approach for systems with severe nonlinearity such as the chip refiner motor load control problem and the pH control problem. Recently, Finn, Wahlberg and Ydstie [27] reformulated Dynamic Matrix Control (DMC) based on a discrete-time Laguerre expansion. In this section, we propose the SDGPC algorithm based on continuous-time Laguerre filter modelling. It is shown that the resulting control law is particularly suitable for adaptive control applications in that its computation burden is independent on the prediction horizon used in the SDGPC design.

Define $x^{\mathbf{T}}(t) = [x_1(t) \quad x_2(t) \quad \dots \quad x_N(t)\,]$. From Fig.4.32, we have the following state space expression of the Laguerre network:

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t) \\ y(t) &= c^T x(t) \end{aligned} \tag{4.164}$$

where

$$A = \begin{bmatrix} -p & 0 & \dots & 0 \\ -2p & -p & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -2p & -2p & \dots & -p \end{bmatrix}_{N\times N} \tag{4.165}$$

$$\begin{aligned} B &= \begin{bmatrix} \sqrt{2p} \\ \sqrt{2p} \\ \vdots \\ \sqrt{2p} \end{bmatrix}_{N\times 1} \\ c^T &= [c_1 \ c_2 \quad \cdots c_N]_{1\times N} \end{aligned} \tag{4.166}$$

With the time scale p being properly selected, the Laguerre spectrum $c^T = [c_1 \quad c_2 \quad \cdots \quad c_N\,]$ can be estimated based on the input output of the process. In fact, notice that the equation $y(t) = c^T x(t)$ in (4.164) is already in a regression model form, various Recursive Least Squares algorithm can be applied. We settled for the recent EFRA (Exponential Forgetting and Resetting Algorithm) [68] which will be described in chapter 6, section 6.1.

Now the SDGPC design procedure can be readily applied to the Laguerre state space equation (4.164). The intended application of Laguerre filter based SDGPC is adaptive control where the Laguerre coefficients $c^T = [c_1 \ c_2 \ \cdots c_N]_{1\times N}$ are estimated on-line using EFRA [68]. We show in the following that the on-line computation burden is actually independent on the prediction horizon T_p.

The main computation involved in the calculation of (2.16) is integration. For example the integration $\int_0^{T_p} \Gamma_o^T(T)cc^T\Gamma_o(T)dT$ in K_1. By some simple manipulations, we have:

$$\int_0^{T_p} \Gamma_o^T(T)cc^T\Gamma_o(T)dT = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & c^T \int_0^{T_p} \gamma_i\gamma_j^T dTc & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}_{N_u\times N_u} \tag{4.167}$$

where γ_i, γ_j are the ith and jth column of $\Gamma_o(T) = [\gamma_1 \ \gamma_2 \cdots \gamma_{N_u}]_{n\times N_u}$. The integral $\int_0^{T_p} \gamma_i\gamma_j^T dT$ can be computed off-line and stored. The other integrals in the calculation of (2.16) can be treated similarly so that the on-line computation burden of the control law (2.15) is independent on the prediction horizon T_p.

Although Laguerre filter modeling is known for its capability of dealing with dead time, however for long time-delay, it requires a large number of Laguerre filters causing problems such as slow convergence rate in the Laguerre coefficients estimation and increasing the control law computation burden. An effective way of dealing with this problem is to use the delayed control $u(t-\tau_d)$ as the input to the Laguerre network in Fig.4.32 resulting in a system model:

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t-\tau_d) \\ y(t) &= c^T x(t) \end{aligned} \tag{4.168}$$

Where τ_d can be either estimated or just a rough guess based on prior knowledge about the process. Only the uncertainty of the time-delay needs to be taken care of by Laguerre network modeling. Based on system model (4.168), the SDGPC for time-delay systems in Section can be applied.

Example 4:

Adaptive Laguerre based SDGPC is shown in this example. The plant is given by

$$G_4(s) = e^{-sT}\frac{0.5s-1}{(s+1)^2} \tag{4.169}$$

Where the dead time T varies from $5.5s$ to $4.5s$ as illustrated in Fig.4.33. Four Laguerre filters with pole $p = 1.2$ are used. The delay is assumed to be $5s$ in the design. Thus the system model can be described as

$$\begin{aligned}\dot{x}(t) &= \begin{bmatrix} -1.2 & 0 & 0 & 0 \\ -2.4 & -1.2 & 0 & 0 \\ -2.4 & -2.4 & -1.2 & 0 \\ -2.4 & -2.4 & -2.4 & -1.2 \end{bmatrix} x(t) + \begin{bmatrix} 1.5492 \\ 1.5492 \\ 1.5492 \\ 1.5492 \end{bmatrix} u(t-5) \\ y(t) &= c^T x(t)\end{aligned} \tag{4.170}$$

where c^T is the Laguerre filter coefficients vector which will be estimated using RLS. The algorithm developed in section 4.1 will be used. The design parameters are:

$$\begin{aligned} \tau_d &= 5s \\ T_{exe} &= 0.25s \\ T_p &= 11s \\ N_u &= 6 \\ \lambda &= 1 \\ \gamma &= 1000 \end{aligned} \tag{4.171}$$

where γ is the end point states weighting.

Fig.4.33 shows the plant output and the reference, the control input to the plant and the control derivative , the estimated Laguerre coefficients. Between 0 to 80 seconds, the dead time is 5.5 seconds, between 80 to 160 seconds T is changing to 5 seconds and after 160 seconds it is reduced to 4.5 seconds. As can be seen, the performances are very good in all three cases. By using the prior delay knowledge, less Laguerre filters can be used resulting in quick convergence in the coefficients estimation and thus less transients in the response.

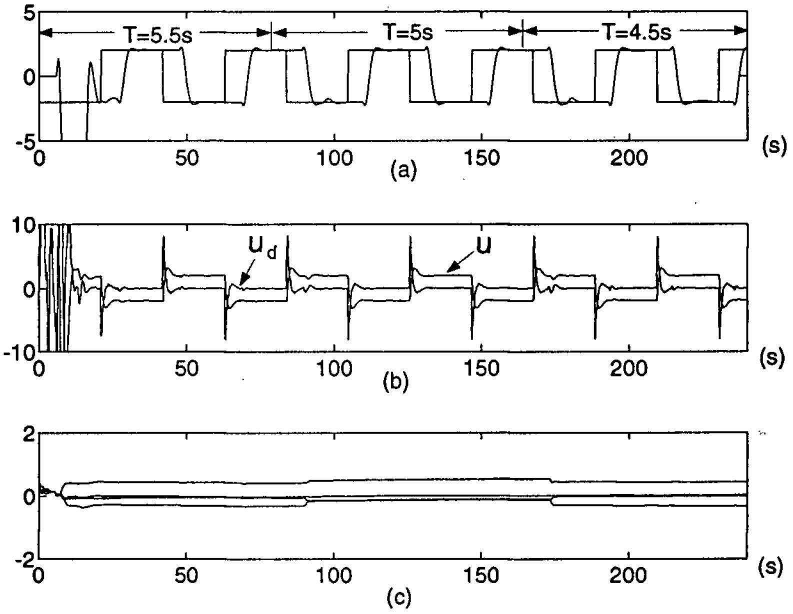

Figure 4.33: Simulation of plant (4.169)

The simulation was performed with the same conditions as the previous run except measurement noise was added at the output. Fig.4.34 shows the system output, setpoint and the estimated Laguerre parameters. The algorithm still performs well in the presence of noise.

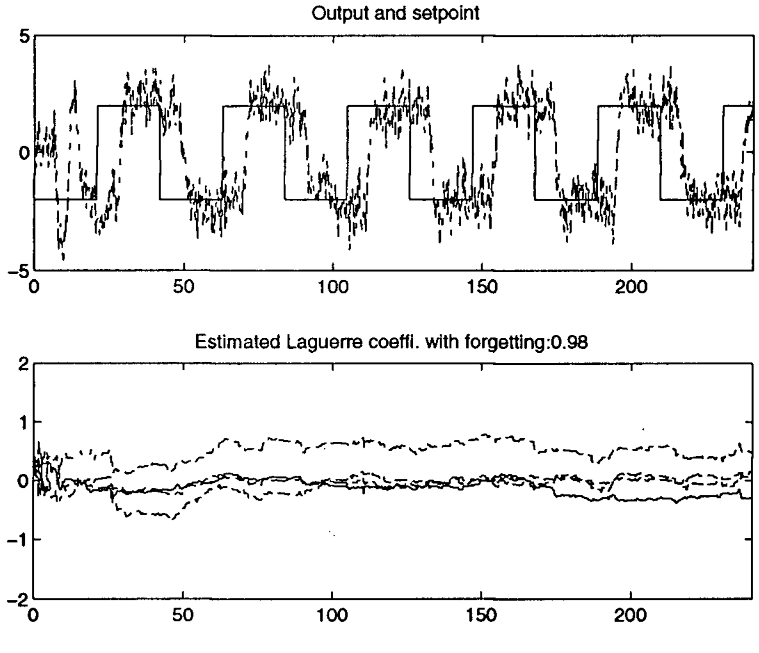

Figure 4.34: Simulation of plant (4.169) with measurement noise

4.3 Conclusion

Model based predictive control can deal with time-delay systems effectively and SDGPC is no exception. Laguerre network requires little *a priori* information about the system and is able to model varying dead times. The adaptive Laguerre filter based SDGPC developed in section 4.2 is a suitable candidate for most process control applications.

Chapter 5

Anti-windup Design of SDGPC by Optimizing Two Performance Indices

Actuation limits exist in almost all practical control systems. For example, a motor has limited speed, a valve can only operate between fully open and fully close etc. Other than these physical actuator limitations, there are constraints which are imposed by production requirements. On the other hand, most of the available controller design methods ignore the existence of the saturation nonlinearity. When large disturbances occur or, the operating conditions are changed over a wide range, it may happen that the theoretical controller output goes beyond the actuator limits. As a result, the control system is effectively operating in open-loop as the input to the plant remains at its limit regardless of the controller output. When this happens, the controller output is wrongly updated. For example, if the controller has integral action, the error will continue to be integrated resulting in a very large integral term. This effect is called *controller windup.* The windup difficulty was first experienced in the integrator part of *PID* controllers. It was recognized later that integrator windup is only a special case of a more general problem. In fact, any controller with relatively slow or unstable modes will experience windup problems if there are actuator constraints [20]. The consequence of windup is either significant performance deterioration, large overshoots or sometimes even instability [8]. Various compensation schemes have been proposed. The *anti-reset windup* method was proposed by Fertik and Ross [26]. Åström and Wittenmark [63 pp. 184–185] suggested resetting the integrator at saturation to prevent integrator windup for PID controllers. A general approach where an observer is introduced into the controller structure to prevent windup was proposed by Åström and Wittenmark [63 pp. 369–373]. The " conditioning technique" was proposed by Hanus [36] and it was found that the conditioned controller can be derived as a special case of the observer-based approach [8]. However, as pointed out in [39], many of these schemes are by and large intuition based. Rigorous stability analyses are rare and there is no general analysis and synthesis theory. Several attempts have been made to unify anti-windup schemes notably by Walgama and Sternby [81] and Kothare et al. [39].

Since the SDGPC control law (2.17) has integral action it will also encounter windup difficulties in the case of actuator saturation unless measures are taken. A systematic approach that takes into account the input constraints right from the start of the problem formulation is the constrained model based predictive control [69]. This is in fact one of the main advantages of using model based predictive control strategy. However, one difficulty which comes with this approach is the increased complexity. The more common approach which most of the afore mentioned schemes adopt is the two steps paradigm. That is, *the linear controller is designed ignoring control input nonlinearities first and then anti-windup algorithm is added to compensate for the performance degradation due to constraints.* This will also be the method of attack used here.

This chapter is organized as follows, in section 5.1, a SDGPC algorithm based on two performance indices is given in which the "nominal" response of the closed-loop system and the integral compensation performance can be designed independently. This was motivated by the work reported in [1, 34, 35, 30] , in which a control algorithm with the structure of state feedback plus integral action was developed where the state feedback and the integral feedback gain can be tuned separately. However, that work was in the framework of continuous-time infinite horizon LQ control. The SDGPC approach, however, has a quite different formulation procedure and an interpretation which naturally leads to a novel anti-windup compensation scheme presented in section 5.2. The importance of this anti-windup scheme is that under some reasonable assumptions, the overall "two-degree-of-freedom" SDGPC and the anti-windup scheme guarantee closed-loop stability. Section 5.3 concludes the chapter.

5.1 SDGPC Based on Two Performance Indices

The primary goal of introducing integral action in the design of SDGPC is to ensure zero static error for systems tracking a non-zero constant setpoint subject to constant disturbances and modeling error to some degree. If there are neither modeling error nor disturbances, there is no need to introduce an additional integral state. However, models are inevitably wrong and there are always disturbances acting on the plant thus integral action is always needed. Nevertheless, the argument is that the controller can be designed for good servo step response performance assuming perfect modeling and no disturbance. Integral action, on the other hand can be added on to compensate for step and impulse disturbances and for modeling error. The key idea is to have the servo performance and disturbance rejection performance tuned independently. In other words, changing the servo performance shall not affect the disturbance rejection performance or vice versa.

5.1.1 Optimizing Servo Performance

Consider system (2.1)

$$\begin{aligned} \dot{x}(t) &= Ax(t)+Bu(t) \\ y(t) &= c^T x(t) \\ dim(x) &= n \end{aligned} \tag{5.172}$$

The system is required to track a constant setpoint r_0. If there is no system zero at the origin, a constant u_0 can be found for any r_0 to hold the system state at x_0 such that [41]:

$$\begin{aligned} 0 &= Ax_0 + Bu_0 \\ y_0 &= c^T x_0 = r_0 \end{aligned} \tag{5.173}$$

Define the *shifted input*, the *shifted state* and the *shifted output* respectively as

$$\begin{aligned} \tilde{u}(t) &= u(t) - u_0 \\ \tilde{x}(t) &= x(t) - x_0 \\ \tilde{y}(t) &= y(t) - r_0 \end{aligned} \tag{5.174}$$

Substitute (5.174) into (5.172) and using (5.173), the shifted variables satisfy the equations

$$\begin{aligned} \dot{\tilde{x}}(t) &= A\tilde{x}(t)+B\tilde{u}(t) \\ \tilde{y}(t) &= c^T \tilde{x}(t) \\ dim(\tilde{x}) &= n \end{aligned} \tag{5.175}$$

A sensible control objective for the system (5.175) is to minimize the following performance index

$$J(t) = \gamma_s \tilde{x}^T(t+T_p)\tilde{x}(t+T_p) + \int_0^{T_p} \left\{ [\tilde{y}(t+T)]^2 + \lambda_s[\tilde{u}(t+T)]^2 \right\} dT \tag{5.176}$$

where $\tilde{u}(\tau), \tau \in [t, t+T_p]$ is confined to be piecewise constant as $u_d(t)$ in (3.76) which is depicted in Fig. 3.17. Follow the same arguments as that of servo SDGPC in chapter 3, section 3.1, the optimal control vector $\tilde{\mathrm{u}}^*$ which minimizes (5.176) can be written as

$$\tilde{\mathrm{u}}^* = K_{\tilde{x}}\tilde{x}(t) \tag{5.177}$$

with

$$\begin{aligned} K_{\tilde{x}} &= K H_{\tilde{x}} \\ K &= \left(\int_0^{T_p} \Gamma^T(T) c c^T \Gamma(T) dT + \lambda_s \int_0^{T_p} H^T(T) H(T) dT + \gamma_s \Gamma^T(T_p)\Gamma(T_p) \right)^{-1}_{N_u \times N_u} \\ H_{\tilde{x}} &= -\left(\int_0^{T_p} \Gamma^T(T) c c^T e^{AT} dT + \gamma_s \Gamma^T(T_p) e^{AT_p} \right)_{N_u \times n} \end{aligned} \tag{5.178}$$

where $\Gamma(T)$ is a $n \times N_u$ matrix given by (2.11) and $H(T)$ is given by (2.7). N_u is the control order as defined before.

Since only the first element of $\tilde{\mathrm{u}}^*$ is applied to the plant according to receding horizon strategy, define F as the first row of the $N_u \times n$ feedback matrix $K_{\tilde{x}}$

$$F = K_{\tilde{x}}(1, 1:n) \tag{5.179}$$

The time-invariant control law for the shifted system is thus

$$\tilde{u}(t) = F\tilde{x}(t) \tag{5.180}$$

Control law (5.180) has guaranteed stability when applied to (5.175) with sampling interval T_m according to *Theorem* 2.3. When the design sampling interval T_m is small, the continuous-time control law also stabilizes the system as shown by simulation in chapter 2, section 2.4. This can be thought of as a procedure of designing continuous-time control law based on sampled-data control in a reverse way of the conventional approach of designing discrete-time control law based on continuous-time design method. For the sake of clarity, (5.180) will be applied to (5.175) with $T_{exe} \rightarrow 0$. This way, the spirit of the anti-windup scheme which will be given in the next section can be shown more clearly and comparisons can be made with that of the familiar three term *PID* control law.

In terms of the original system variables, control law (5.180) can be written from (5.174):

$$u(t) = Fx(t) + u_0 - Fx_0 \tag{5.181}$$

It is easy to see from (5.181) that there is a constant term $u_0' = u_0 - Fx_0$ in the control law. It can be shown [41pp.270-276] that

$$u_0' = \left[c^T\left(-\bar{A}\right)^{-1}B\right]^{-1} r_0 \tag{5.182}$$

where $\bar{A}$ is the closed-loop system matrix

$$\bar{A} = A + BF \tag{5.183}$$

The term $c^T\left(-\bar{A}\right)^{-1}B$ in (5.182) is the static gain of the close-loop system with transfer function

$$H_c(s) = c^T\left(sI - \bar{A}\right)^{-1}B \tag{5.184}$$

from the constant term u_0' to the output. That is

$$H_c(0) = c^T\left(-\bar{A}\right)^{-1}B \tag{5.185}$$

The nonexistence of system zero at the origin ensures that $H_c(0)$ is nonsingular and thus guarantees the existence of such a constant term u_0' given by (5.182) which makes (5.173) hold at steady state. The transfer function from the constant setpoint r_0 to the output is

$$\frac{y(s)}{r_0(s)} = H_c^{-1}(0)c^T\left(sI - \bar{A}\right)^{-1}B = H_c^{-1}(0)H_c(s) \tag{5.186}$$

where the steady state gain is one.

Thus the optimal control law without integral action for (5.172) is

$$u(t) = Fx(t) + H_c^{-1}(0)r_0 \tag{5.187}$$

The control law (5.187) is not "robust" in the sense that when there are either disturbances or modeling error, the output of system (5.172) at steady state will differ from the setpoint r_0. This is where integral control given in the next subsection kicks in.

5.1.2 Optimizing Disturbance Rejection Performance

Define the integral state $z(t)$:

$$z(t) = \int^{t} (r_0 - y(\tau))d\tau \tag{5.188}$$

The integral state augmented system of (5.172) is

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \begin{bmatrix} A & \mathbf{0} \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r_0 \\ y(t) &= [c^T \quad 0] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \end{aligned} \tag{5.189}$$

Compare with (2.3) where the integrator was inserted *before* the plant, the integrator in (5.189) is added after the plant. See. Fig.5.35.

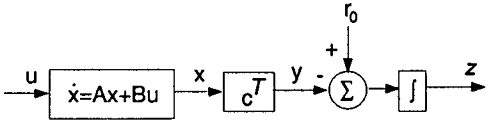

Figure 5.35: Graphical illustration of (5.189)

The last term $[0 \quad 1]^T r_0$ in (5.189) can be ignored since the control law for (5.189) will have integral term which will eliminate any constant disturbance like $[0 \quad 1]^T r_0$. Thus we will work on the disturbance rejection control law based on the following equations

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \begin{bmatrix} A & \mathbf{0} \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) \\ y(t) &= [c^T \quad 0] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \end{aligned} \tag{5.190}$$

What we are looking for is a control law that consists of two terms $u_n(t), v(t)$. The first term $u_n(t)$ is the nominal control given by (5.187) which ensures nominal servo performance. The second term $v(t)$ is responsible for disturbance rejection and modeling error compensation. That is

$$\begin{aligned} u(t) &= u_n(t) + v(t) \\ &= Fx(t) + H_c^{-1}(0)r_0 + v(t) \end{aligned} \tag{5.191}$$

Substitute (5.191) into (5.190), we have

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A & \mathbf{0} \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \\ \begin{bmatrix} B \\ 0 \end{bmatrix} \left(\begin{bmatrix} F & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + H_c^{-1}(0)r_0 \right) + \begin{bmatrix} B \\ 0 \end{bmatrix} v(t) \end{aligned} \tag{5.192}$$

Again ignoring the constant disturbance term $[B \quad 0]^T H_c^{-1}(0)r_0$, we have from (5.192):

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A + BF & \mathbf{0} \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} v(t) \tag{5.193}$$

The disturbance rejection state feedback control law for (5.193) will have the form of

$$v(t) = [L_1 \quad L_2] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \tag{5.194}$$

Without loss of generality, assume $L_2 = \xi, L_1 = \xi L$. i.e.

$$v(t) = \xi[L \quad 1] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \tag{5.195}$$

The closed-loop system of (5.193) under control (5.195) is

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A + BF + \xi BL & \xi B \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \tag{5.196}$$

The criteria for the disturbance rejection control (5.195) are that

1. It should not alter the nominal servo performance given by (5.187). i.e. the eigenvalue of the closed loop system (5.196) should contain the eigenvalues of $\bar{A}$ given by (5.183)
2. It should at the same time give good disturbance rejection properties

Defining the new state $\zeta(t)$ as

$$\zeta(t) = Lx(t) + z(t) \tag{5.197}$$

and substituting (5.197) into (5.193), the new state equation is

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} = \begin{bmatrix} A + BF & 0 \\ L(A + BF) - c^T & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix} + \begin{bmatrix} B \\ LB \end{bmatrix} v(t) \tag{5.198}$$

Substitute control law (5.195), where ξ, L are yet to be decided, into (5.198). The closed loop system is

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} = \begin{bmatrix} A + BF & \xi B \\ L(A + BF) - c^T & \xi LB \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix} \tag{5.199}$$

To fulfill criterion 1, it is obvious that $L(A + BF) - c^T$ has to be a zero vector. i.e. L should be given by

$$L = c^T(A + BF)^{-1} \tag{5.200}$$

Since $(A + BF)$ is the nominal closed loop system matrix which is stable, it is always possible to find L from (5.200). The closed loop system equation is thus

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} = \begin{bmatrix} A + BF & \xi B \\ 0 & \xi LB \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix} \tag{5.201}$$

The eigenvalues of (5.201) are the solutions of

$$det \begin{bmatrix} sI_n - (A + BF) & -\xi B \\ 0 & s - \xi LB \end{bmatrix} = det[sI_n - (A + BF)](s - \xi LB) = 0 \tag{5.202}$$

That is, the eigenvalues of (5.201) are those of $(A + BF)$ and one at ξLB. Given the desired pole location p_z, the feedback gain ξ can be easily decided by

$$\xi = p_z(LB)^{-1} \tag{5.203}$$

The gain ξ can also be given by applying SDGPC method to the following equation

$$\dot{\zeta}(t) = LBv(t) \tag{5.204}$$

The performance index is

$$J = \gamma_r \zeta^2(t+T_p) + \int_0^{T_p} \left[\zeta^2(t+T) + \lambda_r v^2(t+T)\right] dT \tag{5.205}$$

Applying the formulae (5.178) with design parameters

$$\begin{aligned} N_u &= 1 \\ \lambda_r &= 0 \\ \gamma_r &= 0 \end{aligned} \tag{5.206}$$

to (5.205), we have

$$\xi = -\frac{1.5}{T_p}(LB)^{-1} \tag{5.207}$$

The closed loop pole is at $p_z = -\frac{1.5}{T_p}$.

The overall control law (5.191) which takes into account both servo performance and disturbance rejection performance is thus

$$u(t) = (F + \xi L)x(t) + \xi z(t) + K_r r_0 \tag{5.208}$$

Where F is given by (5.179) and

$$\begin{aligned} L &= c^T(A+BF)^{-1} \\ \xi &= -\frac{1.5}{T_p}(LB)^{-1} \\ K_r = H_c^{-1}(0) &= -\left[c^T(A+BF)^{-1}B\right]^{-1} = -(LB)^{-1} \end{aligned} \tag{5.209}$$

We summarize the above results in the following theorem.

***Theorem* 5.1**

For the system given by (5.172) and the augmented system (5.190)

1. *The eigenvalues of the closed loop system under control law (5.208) are those of $A + BF$ and $p_z = \xi(LB)$. Where F and ξ are obtained by minimizing two performance indices (5.176) and (5.205) respectively using SDGPC method. L is given by (5.209).*
2. *The system transfer function from reference r_0 to system output $y(t)$ under control law (5.208) is the same as that of the nominal case (5.186). i.e. $K_r c^T(sI_n - (A + BF))^{-1}B$.*

Proof:

1. The closed loop system equations of (5.190) under control law (5.208) are:

$$\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\end{bmatrix} = \begin{bmatrix}A + BF + \xi BL & B\xi\\ -c^T & 0\end{bmatrix}\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} + \begin{bmatrix}B\\ 0\end{bmatrix}K_r r_0$$
$$y(t) = [c^T \quad 0]\begin{bmatrix}x(t)\\ z(t)\end{bmatrix} \tag{5.210}$$

Apply the similarity transformation defined by nonsingular matrix $T = \begin{bmatrix}I_n & 0\\ -L & 1\end{bmatrix}$ to the closed loop system matrix

$$A_{close} = \begin{bmatrix}A + BF + \xi BL & B\xi\\ -c^T & 0\end{bmatrix} \tag{5.211}$$

$$\hat{A}_{close} = T^{-1}A_{close}T$$
$$= \begin{bmatrix}A + BF + \xi BL - \xi BL & \xi B\\ L(A + BF) - c^T + \xi LBL - \xi LBL & \xi LB\end{bmatrix} \tag{5.212}$$
$$= \begin{bmatrix}A + BF & \xi B\\ 0 & \xi LB\end{bmatrix}$$

The eigenvalues of $\hat{A}_{close}$ are given by

$$det\left[sI_{n+1} - \begin{bmatrix}A + BF & \xi B\\ 0 & \xi LB\end{bmatrix}\right] =$$
$$det\begin{bmatrix}sI_n - (A + BF) & -\xi B\\ 0 & s - \xi LB\end{bmatrix} = \tag{5.213}$$
$$det[sI_n - (A + BF)]det(s - \xi LB) = 0$$

So the eigenvalues of $\hat{A}_{close}$ are eigenvalues of $A+BF$ and ξLB. Since similarity transformation does not change the eigenvalues of a matrix, it is clear that the eigenvalues of $A_{close} = \begin{bmatrix} A+BF+\xi BL & B\xi \\ -c^T & 0 \end{bmatrix}$ are those of $\hat{A}_{close}$. This completes the proof of statement 1.

2. Substitute control law (5.208) into system equations (5.189), we have

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \begin{bmatrix} A+BF+\xi BL & B\xi \\ -c^T & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} BK_r \\ 1 \end{bmatrix} r_0 \\ y(t) &= [c^T \quad 0] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \end{aligned} \tag{5.214}$$

Apply transformation $\begin{bmatrix} x(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ -L & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix}$ to (5.214), we have

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} &= \begin{bmatrix} A+BF & B\xi \\ 0 & LB\xi \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix} + \begin{bmatrix} BK_r \\ LBKr+1 \end{bmatrix} r_0 \\ y(t) &= [c^T \quad 0] \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \end{aligned} \tag{5.215}$$

Using (5.209), we have $LBK_r + 1 = 0$, the closed loop transfer function from r_0 to $y(t)$ is thus

$$\begin{aligned} & [c^T \quad 0] \begin{bmatrix} sI_n - (A+BF) & -B\xi \\ 0 & s - LB\xi \end{bmatrix}^{-1} \begin{bmatrix} BK_r \\ 0 \end{bmatrix} \\ = [c^T \quad 0] & \left\{ \begin{matrix} [sI_n - (A+BF)]^{-1} & [sI_n - (A+BF)]^{-1} B\xi (s - LB\xi)^{-1} \\ 0 & (s - LB\xi)^{-1} \end{matrix} \right\} \begin{bmatrix} BK_r \\ 0 \end{bmatrix} \\ & = c^T [sI_n - (A+BF)]^{-1} BK_r \end{aligned} \tag{5.216}$$

This completes the proof of statement 2. □

Remark:

Statement 2 implies that the integral term in control law (5.208) adds a systems pole as well as a system zero at ξLB. However, the integral term adds a blocking zero at the origin from the reference (or disturbance) to the tracking error $e(t) = r_0 - y(t)$ thus ensures zero steady state error for constant reference signal and/or constant disturbance.

Theorem 5.1 says for a changing ξ, ξ_t, the eigenvalues of the closed loop system are those of $A+BF$, which are constant, and a time-varying pole at $p_{z_t} = \xi_t LB$. For stable $A+BF$ and a stable p_z given by ξ, as long as the sign of ξ_t stays the same as ξ, then the eigenvalues of the closed loop system matrix (5.211) are stable at any instant of time. However, this does not guarantee the stability of the time-varying $A^t_{close} = \begin{bmatrix} A+BF+\xi_t BL & B\xi_t \\ -c^T & 0 \end{bmatrix}$. That is, for time-varying $\dot{x}(t) = A(t)x(t)$, if the eigenvalues of $A(t)$ have real negative parts for each t, then $\dot{x}(t) = A(t)x(t)$ is *not* necessarily stable [9, pp.411]. For the particular $A^t_{close} = \begin{bmatrix} A+BF+\xi_t BL & B\xi_t \\ -c^T & 0 \end{bmatrix}$, we have the following theorem.

***Theorem* 5.2**

For the system (5.172) and the augmented system (5.190) under control law (5.208) with constant setpoint and with constant F, which stabilizes $A+BF$, and time-varying ξ_t, if

1. *ξ_t is such that the time-varying pole $p_{z_t} = \xi_t LB$ is always negative, i.e. $p_{z_t} = \xi_t LB < 0$,* ***and***
2. *$\xi_t(t)e^{\int_0^t LB\xi_t(\tau)d\tau}$ is bounded* ***and*** *$\lim_{t\to\infty} \xi_t(t)e^{\int_0^t LB\xi_t(\tau)d\tau}$ = constant exponentially,*

then the closed loop system (5.214) is exponentially stable.

Proof: For the closed loop system

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} = \begin{bmatrix} A+BF & B\xi_t(t) \\ 0 & LB\xi_t(t) \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix} \tag{5.217}$$

we have

$$\begin{aligned} \frac{d\zeta}{\zeta} &= LB\xi_t(t)dt \\ \zeta(t) &= \zeta(0)e^{\int_0^t LB\xi_t(\tau)d\tau} \end{aligned} \tag{5.218}$$

Condition *1.* ensures $\lim_{t\to\infty} \int_0^t LB\xi_t(\tau)d\tau$ is a negative constant (not necessarily $-\infty$) which means

$$\lim_{t\to\infty} \zeta(t) = constant \tag{5.219}$$

from (5.218). For the states $x(t)$, we have

$$\begin{aligned} \dot{x}(t) &= (A+BF)x(t) + B\xi_t(t)\zeta(t) \\ x(t) &= e^{(A+BF)t}x(0) + \int_0^t e^{(A+BF)(t-\tau)}B\xi_t(\tau)\zeta(\tau)d\tau \end{aligned} \tag{5.220}$$

Because $A + BF$ is stable and $\lim_{t\to\infty} \xi_t(t)\zeta(t) = constant\,(or\ 0)$ according to condition 2., which means we have a stable system subject to an input which is either exponentially vanishing or goes to a nonzero constant exponentially. Thus

$$\lim_{t\to\infty} x(t) = x_\infty\,(constant) \tag{5.221}$$

i.e. exponentially stable at its equilibrium point. □

Theorem 5.2 will be used to construct an anti-windup scheme in next subsection.

In the process industries, most processes are well damped open-loop stable systems. For such systems, the " mean level " control [13], i.e. the control law which does not alter the process poles, may be desirable since more aggressive control requires large input amplitude. The following theorem gives the SDGPC version of " mean level " control.

***Theorem* 5.3**

For system (5.172) with stable A and the augmented system (5.190), the mean level control law (5.223), i.e. the one with $F = \mathbf{0}$, *is obtained with design parameters*

$$\begin{aligned} N_u &= 1 \\ T_p &= \infty \\ \lambda &= 0 \\ \gamma &= 0 \end{aligned} \tag{5.222}$$

The mean level control law is:

$$u(t) = (\xi L)x(t) + \xi z(t) + K_r r_0 \tag{5.223}$$

where ξ *is given by (5.207) and* $K_r = -\left[c^T A^{-1} B\right]^{-1}$.

Proof:

For stable A and $N_u = 1$, $\Gamma(T)$ in (5.178) has a concise form:

$$\Gamma(A,T) = \left(e^{AT} - I\right)A^{-1}B \tag{5.224}$$

Substitute (5.224) into (5.178), we have

$$K = \left(\int_0^{T_p} \left[\left(c^T e^{AT} A^{-1} B\right)^2 - 2c^T e^{AT} A^{-1} B c^T A^{-1} B + \left(c^T A^{-1} B\right)^2 \right] dT \right)^{-1} \qquad (5.225)$$
$$= (I_1 - I_2 + I_3)^{-1}$$

Since A is stable, $\lim_{T_p \to \infty} e^{AT_p} = \mathbf{0}$, thus both the first and the second integrals I_1, I_2 in (5.225) approach constant while I_3 approaches infinity as $T_p \to \infty$. Thus $\lim_{T_p \to \infty} K = 0$. Similarly, it can be shown that $H_{\tilde{x}}$ approaches constant as $T_p \to \infty$. So $\lim_{T_p \to \infty} F = \lim_{T_p \to \infty} K * H_{\tilde{x}} = \mathbf{0}$. □

Remark: Although the mean level control law (5.223) does not change the system dynamics of the setpoint response from the open loop one, its disturbance rejection response can be tuned by selecting different ξ. Larger ξ results in faster disturbance rejection rate.

5.2 Anti-windup Scheme

Fig. 5.36 shows the control law (5.208) applied to a system subject to actuator constraints where

$$\hat{u}(t) = sat[u(t)] = \begin{cases} u(t), & u_{min} \le u(t) \le u_{max} \\ u_{max}, & u_{max} \le u(t) \\ u_{min}, & u(t) \le u_{min} \end{cases} \qquad (5.226)$$

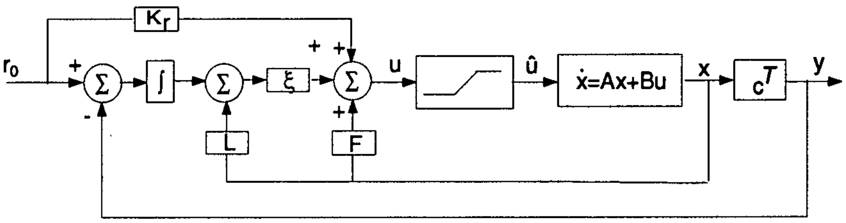

Figure 5.36: Control subject to actuator constraints

As we mentioned earlier, $u(t)$ consists of the nominal control term $u_n(t)$ and the disturbance rejection control term $v(t)$. i.e.

$$u(t) = u_n(t) + v(t)$$
$$u_n(t) = Fx(t) + K_r r_0 \qquad (5.227)$$
$$v(t) = \xi(Lx(t) + z(t))$$

We only consider open loop stable plants here as it is meaningless to design anti-wind up scheme for strictly unstable systems. It is impossible to stabilize a strictly unstable system regardless of whatever control strategy is applied when the system disturbance causes the input to saturate [47].

We first consider the case when $u(t)$ is over the upper limit u_{max}.

*Case 1: Do not reset integrator when **both** $u(t)$ and $u_n(t)$ exceed control limit. That is:*

$$If\begin{cases} u(t) > u_{max} \quad \text{and } u_n(t) > u_{max} \ \mathit{or} \ u_n(t) < u_{min}, \ \mathit{then}\begin{cases} z_{reset}(t)=z(t) \\ \bar{u}(t)=u_{max} \end{cases} \\ u(t) < u_{min} \quad \text{and } u_n(t) < u_{min} \ \mathit{or} \ u_n(t) > u_{max}, \ \mathit{then}\begin{cases} z_{reset}(t)=z(t) \\ \bar{u}(t)=u_{min} \end{cases} \end{cases} \tag{5.228}$$

where $\bar{u}(t)$ is defined as the new controller output after integrator reset.

Case 1 happens either when an unreachable setpoint is asked or the system suffers too large a disturbance. The system is effectively operated in open loop. If the input is saturated long enough, the output of the system will be $-c^T A^{-1} B u_{lim}$.

Case 2: Reset integrator state when $u(t)$ exceeds control limit but $u_n(t)$ does not. Stop reset integrator when saturation is over. That is:

1. *When saturation occurs:*

$$If\begin{cases} u(t) > u_{max} \text{ and } u_{min} < u_n(t) < u_{max}, \quad \mathit{then}\begin{cases} z_{reset}(t) = z(t) + \frac{u_{max}-u(t)}{\xi} \\ \bar{u}(t) = u_n(t) + \xi(Lx(t) + z_{reset}(t)) \end{cases} \\ u(t) < u_{min} \text{ and } u_{min} < u_n(t) < u_{max}, \quad \mathit{then}\begin{cases} z_{reset}(t) = z(t) + \frac{u_{min}-u(t)}{\xi} \\ \bar{u}(t) = u_n(t) + \xi(Lx(t) + z_{reset}(t)) \end{cases} \end{cases} \tag{5.229}$$

2. *After saturation is over, the control should be updated according to:*

$$\begin{aligned} \bar{z}(t) &= z(t) \\ \bar{u}(t) &= u_n(t) + \xi(Lx(t) + \bar{z}(t)) \end{aligned} \tag{5.230}$$

We have the following theorem regarding scheme (5.229).

***Theorem* 5.4**

Scheme (5.229) guarantees $\bar{u}(t) = \hat{u}(t)$ where $\hat{u}(t) = sat[u(t)]$ is defined in (5.226).

Proof:

Consider the case when $u(t) > u_{max}$ *and* $u_{min} < u_n(t) < u_{max}$, by direct calculation we have

$$\begin{aligned}\bar{u}(t) &= u_n(t) + \xi\left(Lx(t) + z(t) + \frac{u_{max} - u(t)}{\xi}\right) \\ &= u_{max} + u_n(t) + Lx(t) + z(t) - z(t) = u_{max}\end{aligned} \tag{5.231}$$

Thus $\hat{u}(t) = sat[u(t)] = \bar{u}(t)$. □.

Remark:

Theorem 5.4 essentially claims that the nonlinear control problem with saturated $u(t)$ can be transformed into a linear control problem by scheme (5.229).

Case 3: Reset the feedback gain ξ but keep the integrator state unchanged when $u(t)$ exceeds control limit but $u_n(t)$ does not. After saturation is over, changing ξ according to equation (5.233). That is:

1. *When saturation occurs:*

$$If \begin{cases} u(t) > u_{max} \text{ and } u_{min} < u_n(t) < u_{max}, & then \begin{cases} \xi_{reset} = \xi + \frac{u_{max} - u(t)}{Lx(t) + z(t)} \\ \bar{u}(t) = u_n(t) + \xi_{reset}(Lx(t) + z(t)) \end{cases} \\ u(t) < u_{min} \text{ and } u_{min} < u_n(t) < u_{max}, & then \begin{cases} \xi_{reset} = \xi + \frac{u_{min} - u(t)}{Lx(t) + z(t)} \\ \bar{u}(t) = u_n(t) + \xi_{reset}(Lx(t) + z(t)) \end{cases} \end{cases} \tag{5.232}$$

2. *After saturation is over, the disturbance feedback gain $\bar{\xi}(t)$ should fulfill the following equation:*

$$\bar{\xi}(t)e^{\int_{t_0}^{t} LB\bar{\xi}(\tau)d\tau} = \xi_{reset}(t_0)e^{\xi LB(t-t_0)} \tag{5.233}$$

where t_0 denotes the time just when saturation is over, ξ is the originally designed feedback gain without considering saturation. $\xi_{reset}(t_0)$ is the reset ξ at t_0 obtained from (5.232).

lemma 5.1

The integrator reset algorithm given by (5.229) and (5.230) is equivalent to the gain reset algorithm given by (5.232) and (5.233).

We only prove the case when the control exceeds the upper limit u_{max}. The case when $u < u_{min}$ can be proved in the same manner.

Proof:

1. When control is saturated. i.e. $u(t) > u_{max}$. From Theorem 5.4, it was shown that (5.229) gives $\bar{u}(t) = u_{max}$. While in (5.232), we have

$$\begin{aligned}\bar{u}(t) &= u_n(t) + \left(\xi + \frac{u_{max} - u(t)}{Lx(t) + z(t)}\right)(Lx(t) + z(t)) \\ &= u_n(t) + \xi(Lx(t) + z(t)) + u_{max} - u(t) \qquad (5.234) \\ &= u_{max}\end{aligned}$$

So, both (5.229) and (5.232) give the same control action when $u(t) > u_{max}$.

2. After saturation is over, denote that moment as t_0, we have

$$\begin{aligned}\bar{u}_1(t_0) &= u_n(t_0) + \xi(Lx(t_0) + z_{reset}(t_0)) = u_n(t_0) + \xi\zeta_{reset}(t_0) \\ \bar{u}_2(t_0) &= u_n(t_0) + \xi_{reset}(t_0)(Lx(t_0) + z(t_0)) = u_n(t_0) + \xi_{reset}(t_0)\zeta(t_0)\end{aligned} \qquad (5.235)$$

where $\bar{u}_1(t_0), \bar{u}_2(t_0)$ are the control obtained from (5.229) and (5.232) respectively. Since $\bar{u}_1(t_0) = \bar{u}_2(t_0)$ as we just proved, we have

$$\xi\zeta_{reset}(t_0) = \xi_{reset}(t_0)\zeta(t_0) \qquad (5.236)$$

The control given by (5.229) after t_0 is

$$\bar{u}_1(t) = u_n(t) + \xi\zeta_{reset}(t) \qquad (5.237)$$

where $\zeta_{reset}(t)$ is given by (5.201) as:

$$\begin{aligned}\dot{\zeta}_{reset}(t) &= LB\xi\zeta_{reset}(t) \\ \zeta_{reset}(t) &= \zeta_{reset}(t_0)e^{\xi LB(t-t_0)} \qquad (5.238) \\ \bar{u}_1(t) &= u_n(t) + \xi\zeta_{reset}(t_0)e^{\xi LB(t-t_0)}\end{aligned}$$

Similarly, the control given by (5.232) after t_0 is

$$\bar{u}_2(t) = u_n(t) + \bar{\xi}(t)\zeta(t) \qquad (5.239)$$

where at time t_0, $\bar{\xi}(t_0) = \xi_{reset}(t_0)$. After t_0, $\bar{\xi}(t)$ is described by (5.233). Finally, we have

$$\begin{aligned}\dot{\zeta}(t) = LB\bar{\xi}(t)\zeta(t), \ \zeta(t) &= \zeta(t_0)e^{\int_{t_0}^{t} LB\bar{\xi}(\tau)d\tau} \\ \bar{u}_2(t) = u_n(t) + \bar{\xi}(t)\zeta(t_0)&e^{\int_{t_0}^{t} LB\bar{\xi}(\tau)d\tau}\end{aligned} \qquad (5.240)$$

For $\bar{\xi}(t)$ given by (5.233), it is easy to show that $\bar{u}_1(t) = \bar{u}_2(t)$ using (5.236).

We conclude that the integral state reset algorithm described by (5.229) and (5.230) and the gain reset algorithm described by (5.232) and (5.233) give the same unsaturated control action during saturation period and thereafter and are thus equivalent. □

Remark: Lemma 5.1 claims that the nonlinear control problem with saturated $u(t)$ is equivalent to a linear time-varying control problem with the time-varying feedback gain given by (5.232) and (5.233).

lemma 5.2

For the gain reset scheme (5.232) and (5.233), the sign of $\xi_{reset}(t)$ during saturation and the sign of $\bar{\xi}(t)$ after saturation are the same as the sign of the originally designed ξ.

Proof:

1. During saturation, from (5.232), we have

$$\xi_{reset}(t) = \xi + \frac{u_{max} - u(t)}{Lx(t) + z(t)} = \frac{u_{max} - u_n(t)}{Lx(t) + z(t)} \tag{5.241}$$

Since $u_{max} - u_n(t) > 0$ by assumption, we have

$$\xi_{reset}(t)(Lx(t) + z(t)) > 0 \tag{5.242}$$

We also have $u(t) = u_n(t) + \xi(Lx(t) + z(t)) > u_{max}$, thus

$$\xi(Lx(t) + z(t)) > u_{max} - u_n(t) > 0 \tag{5.243}$$

Equations (5.242) and (5.243) hold at the same time only if ξ and $\xi_{reset}(t)$ have the same sign.

2. After the saturation, from (5.233), it is obvious that

$$\frac{\bar{\xi}(t)}{\xi_{reset}(t_0)} = \frac{e^{\xi LB(t-t_0)}}{e^{\int_{t_0}^{t} LB\bar{\xi}(\tau)d\tau}} > 0 \tag{5.244}$$

since both $e^{\xi LB(t-t_0)}$, $e^{\int_{t_0}^{t} LB\bar{\xi}(\tau)d\tau}$ are greater than zero. Thus $\bar{\xi}(t)$ has the same sign as $\xi_{reset}(t_0)$.

We conclude that the signs of ξ, $\xi_{reset}(t)$ and $\bar{\xi}(t)$ are all the same. □

***Theorem* 5.5**

The gain reset anti-windup scheme described by (5.232) and (5.233) and the equivalent integrator reset anti-windup scheme described by (5.229) and (5.230) are exponentially stable and gives zero steady state error for constant setpoint subject to constant disturbance.

Proof:

According to lemma 5.2, the time-varying ξ_t has the same sign as ξ. Since $\xi LB < 0$ by design, we have $p_{z_t} = \xi_t LB < 0$ during all the stages of the algorithm. Further ξ_t is bounded during saturation and $\lim_{t\to\infty} \bar{\xi}(t)e^{\int_{t_0}^{t} LB\bar{\xi}(\tau)d\tau} = \lim_{t\to\infty} \xi_{reset}(t_0)e^{\xi LB(t-t_0)} = 0$ after saturation according to (5.233). Thus both condition *1.* and condition *2.* of Theorem 5.2 are met. The resulting time-varying ***linear*** control law is exponentially stable for the overall system, i.e. the linear system (5.172) with saturation nonlinearity as depicted in Fig. 5.36. The equivalent integrator state reset algorithm (5.229) and (5.230) is also exponentially stable. Further, since $\lim_{t\to\infty} \zeta(t) = \lim_{t\to\infty} (Lx(t) + z(t)) = constant$ and $\lim_{t\to\infty} x(t) = constant$, we have $\lim_{t\to\infty} z(t) = \lim_{t\to\infty} \int^{t} (r_0 - y(\tau))d\tau = constant$, which means $\lim_{t\to\infty} e(t) = \lim_{t\to\infty} (r_0 - y(t)) = 0$. □

Some examples are presented here to show the effectiveness of the TDF-SDGPC and the anti-windup algorithm.

Example 5.1 The first example is a simple integrator process.

$$G(s) = \frac{1}{s} \tag{5.245}$$

The actuator limits are ±0.1.

Design the SDGPC algorithm with two performance indices using the same following design parameters

$$\begin{aligned} N_u &= 1 \\ T_p &= 1.5sec \\ \lambda &= \gamma = 0 \end{aligned} \tag{5.246}$$

gives the control law

$$\begin{aligned} u(t) &= u_n(t) + v(t) \\ u_n(t) &= -y(t) + r_0 \\ v(t) &= -y(t) + z(t) \end{aligned} \tag{5.247}$$

Fig.5.37 shows the control results of control law (5.247) without anti-windup compensation. The overshoot in Fig.5.37(a) can be clearly observed.

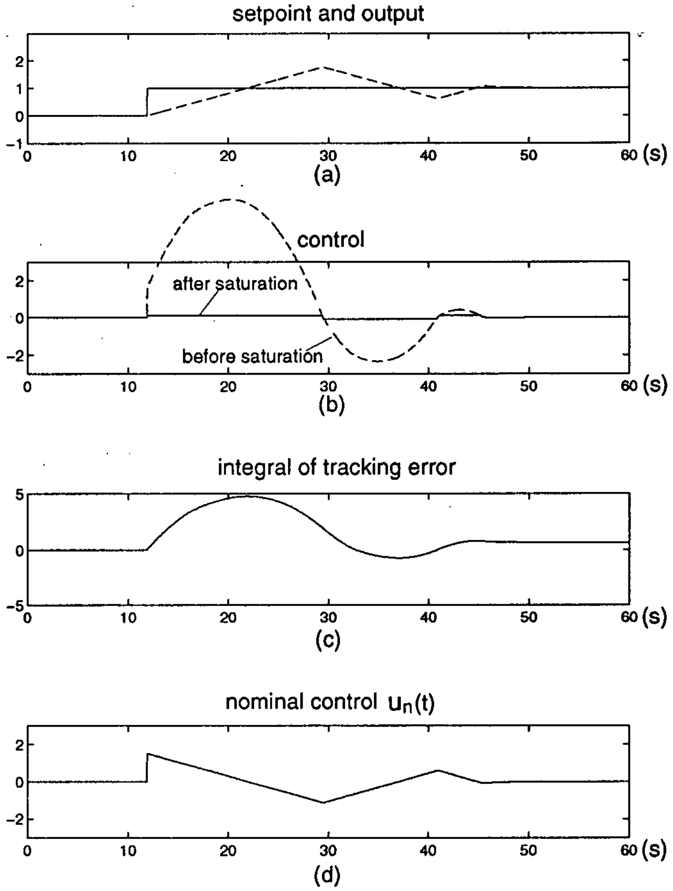

Figure 5.37: Example 5.1: Control law (5.247) without anti-windup compensation

Fig. 5.38 shows the control results of control law (5.247) with anti-windup scheme. From Fig. 5.38(c), the integral state is reset just after the 20 seconds mark at which time the nominal control is within the control limits. The effectiveness of the algorithm is obvious.

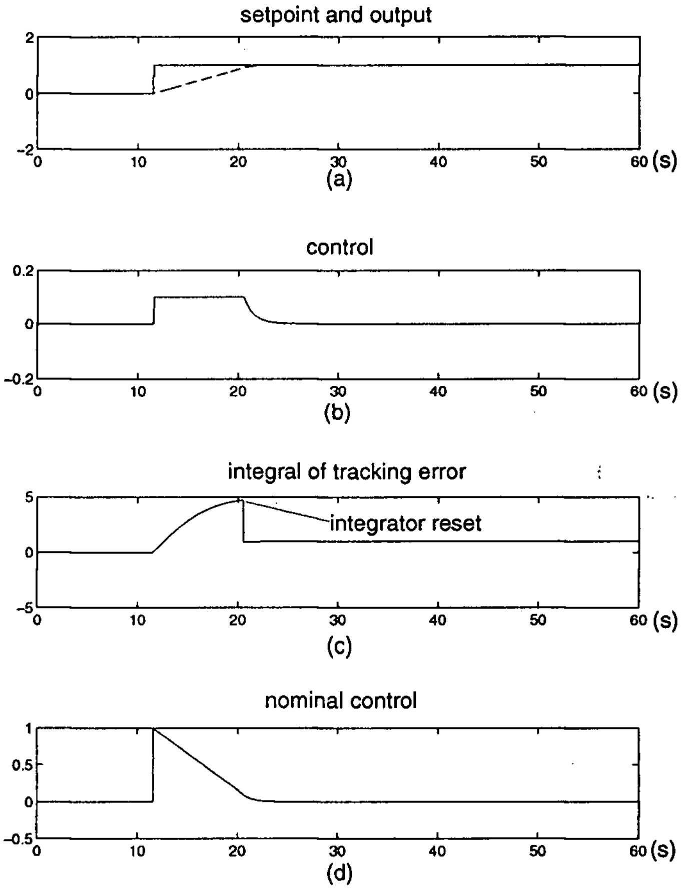

Figure 5.38: Example 5.1: Control law (5.247) with anti-windup compensation

Example 5.2 The process for the second example is described by

$$\dot{x}(t) = \begin{bmatrix} -3 & -3 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(t)$$
$$y(t) = [0 \quad 0 \quad 1]x(t) \tag{5.248}$$

The actuator limits are: ± 3.5

For the nominal control law $u_n(t)$, the design parameters are

$$
\begin{aligned}
N_u &= 10 \\
T_p &= 2sec. \\
\lambda &= 0.0001 \\
\gamma &= 1000
\end{aligned}
\tag{5.249}
$$

The resulting nominal control law is given by

$$
\begin{aligned}
u_n(t) &= Fx(t) + K_r r_0 \\
F &= [-4.7599 \quad -25.9369 \quad -51.4516\,], K_r = 52.4516
\end{aligned}
\tag{5.250}
$$

For the control $v(t)$, the prediction horizon T_p is selected such that the pole is placed at -2 and $\xi = 104.9031$. the overall control law is

$$
\begin{aligned}
u_n(t) &= (F + \xi L)x(t) + K_r r_0 + \xi z(t) \\
&= [-6.7599 \quad -41.4567 \quad -109.3254\,]x(t) + 52.4516 r_0 + 104.9031 z(t)
\end{aligned}
\tag{5.251}
$$

The control results for the control law (5.251) with anti-windup algorithm are shown in Fig. 5.39. The effectiveness of the anti-windup algorithm can be again observed by comparing Fig. 5.39 with Fig. 5.40 in which no anti-windup algorithm is used. Notice the integral state reset in Fig. 5.39(c) after each setpoint change.

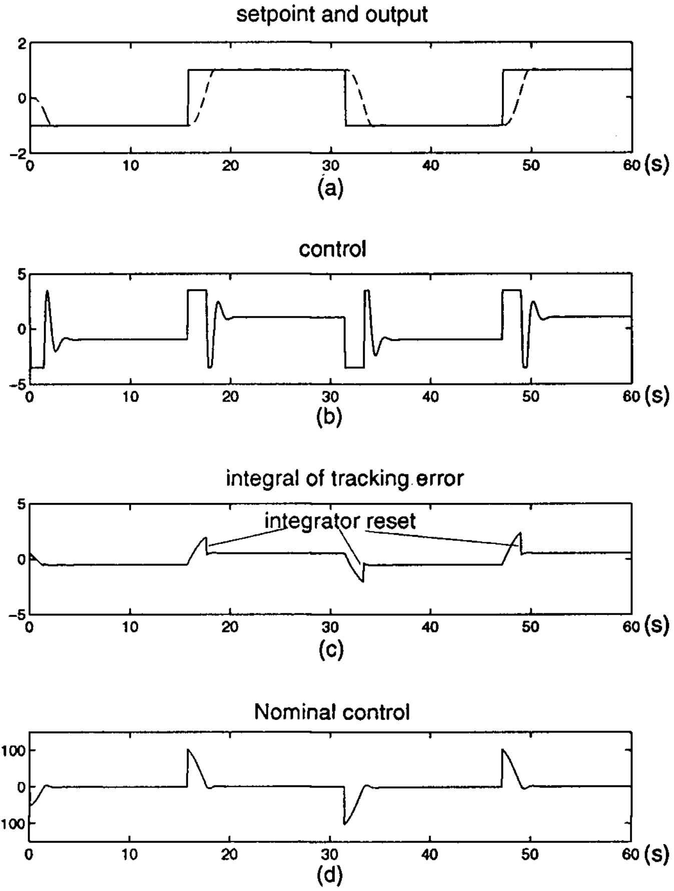

Figure 5.39: Example 5.2: Control law (5.251) with anti-windup compensation

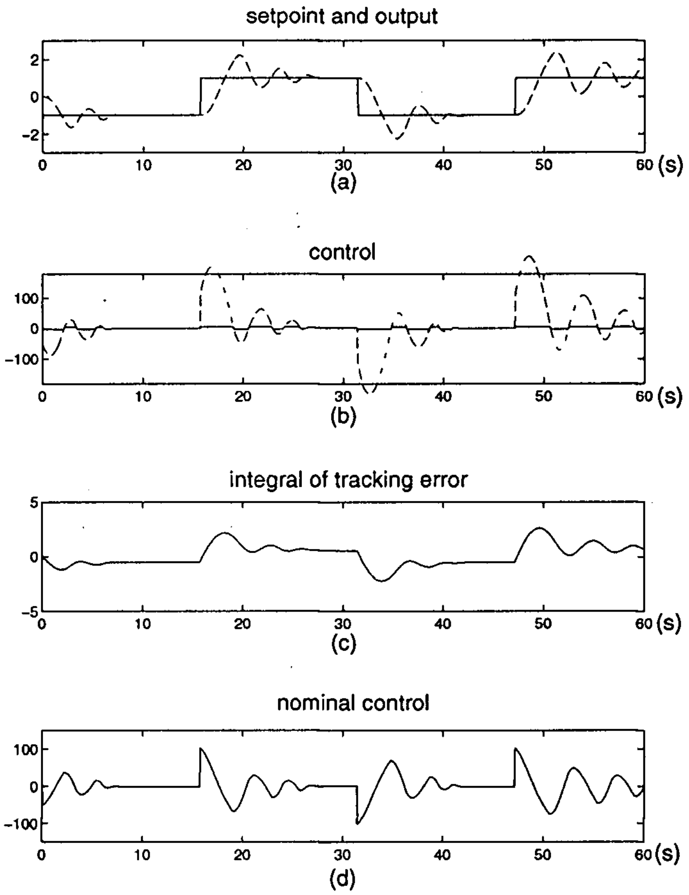

Figure 5.40: Example 5.2: Control law (5.251) without anti-windup compensation

The integrator reset algorithm has a strong similarity with the conventional anti-windup PID controller whose structure is depicted in Fig.5.41.

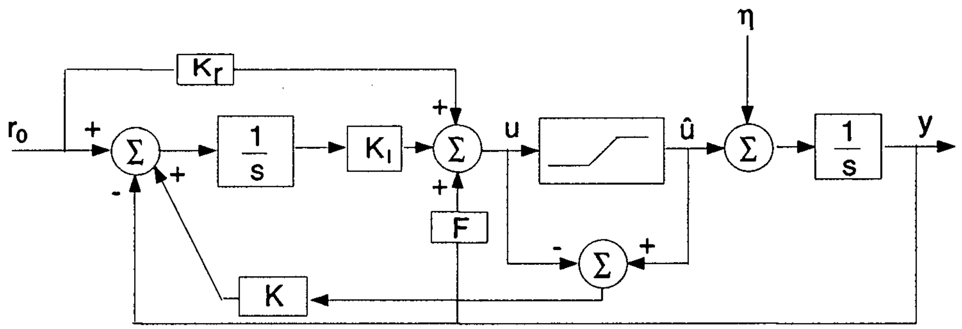

Figure 5.41: Conventional anti-windup

The performances of the conventional anti-windup algorithm and the proposed scheme are compared in next example.

Example 5.3 The plant being controlled is a simple integrator with random disturbance injected at the input. The saturation limits are ± 0.1 and the control law is given in equation (5.252)

$$
\begin{aligned}
G(s) &= \frac{1}{s} \\
-0.1 &\leq u(t) \leq 0.1 \\
u(t) &= -1.9998y(t) + 0.9998r_0 + 0.9998 \int^{t} (r_0 - y(\tau))d\tau \qquad (5.252) \\
u_n(t) &= -0.9998y(t) + 0.9998r_0 \\
\nu(t) &= -y(t) + 0.9998 \int^{t} (r_0 - y(\tau))d\tau
\end{aligned}
$$

The reset gain K as depicted in Fig.5.41 is selected to ensure good performance for the conventional algorithm. Fig.5.42 shows the results. The first three plots are for the conventional methods with different reset gains. The observation is that although good performance can be obtained for properly selected gain K, it is nonetheless a nontrivial trial and error procedure. Moreover, the effect of changing K on the performance is "non-monotonic". This makes the tuning more difficult. On the other hand, the proposed scheme gives excellent results in a straightforward manner as can be seen from the fourth plot of Fig.5.42.

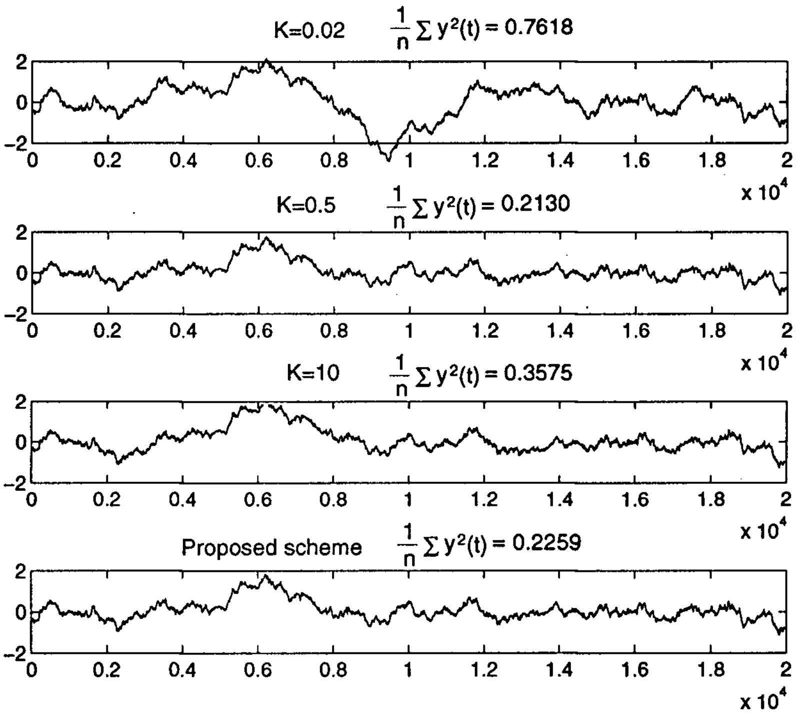

Figure 5.42: Conventional anti-windup scheme vs proposed

We make some remarks to conclude this section.

1. The stability result stated in Theorem 5.5 also applies to unstable systems ***provided*** that $u_{min} < u_n(t) < u_{max}$ holds. However, for unstable plants, it is always possible to find a system state x_0 such that $u_n(t) = Fx_0 + K_r r_0 > u_{max}$ or $u_n(t) < u_{min}$. This will drive the system in open loop mode thus destabilize the system.
2. For open loop stable systems, it is also desirable to have

$$\begin{aligned} u_n(t) &= Fx(t) + K_r r_0 \\ u_{min} < u_n(t) &< u_{max} \end{aligned} \tag{5.253}$$

hold. This way, the anti-windup scheme leads to a graceful performance degradation compared with the unconstrained linear design. However, to fulfill (5.253), a smaller feedback gain F is required. The extreme case is the "mean-level" control given by Theorem 5.3 where $F = \mathbf{0}$, i.e. no control effort is made to make the servo response faster. This means that when input limits exist, a trade-off must be made to balance servo performance and disturbance rejection performance even for a two-degree-of-freedom design.
3. Although the integral state reset scheme (5.229), (5.230) is equivalent to the gain reset scheme (5.232), (5.233), it is clear that the integrator reset scheme is much easier to implement. The gain reset scheme is also very important since it clearly shows that the original time-invariant nonlinear problem is equivalent to a time-varying linear problem which possesses nice stability property under reasonable assumption.
4. Integral reset can also be used for bumpless transfer in process control. For manual control signal $u_{man}(t)$, the integral $z(t)$ in control law (5.208) can be set as

$$z(t) = \frac{u_{man}(t) - (F + \xi L)x(t) - K_r r_0}{\xi} \tag{5.254}$$

This way, the control law (5.208) tracks the operator action in manual mode and takes over the control task from the operator based on the last operator entered value whenever in auto mode thus realizing bumpless transfer.

5.3 Conclusion

In this chapter, we developed a two-degree-of-freedom SDGPC algorithm based on two performance indices. Based on this TDF-SDGPC algorithm, an anti-windup scheme was proposed in section 5.2. It is shown that the linear control law plus saturation nonlinearity is effectively equivalent to an unconstrained linear time-varying control law which leads to graceful performance degradation compared with the original linear unconstrained design while guarantees stability. It also clearly poses a trade-off design problem between the servo performance and the disturbance rejection performance. The attractive features of this anti-windup scheme are its elegant simplicity, effectiveness and most importantly, guaranteed stability property.

Chapter 6

Continuous-time System Identification Based on Sampled-Data

System model is the centerpiece in all Model Based Predictive Control algorithms. In the case of various SDGPC algorithms we developed in previous chapters, the system models being used are continuous-time state space (or high order differential equation) models. The continuous-time Laguerre filter identification and its associated Laguerre filter based adaptive SDGPC problem were treated in chapter 4. The parameter estimation problem for general (stable or unstable) continuous-time differential equation is discussed in this chapter.

There is no doubt that discrete-time models have received more attention than their continuous-time counterparts in the development of both identification theory and techniques. The theoretical results and algorithms available for discrete-time model parameter estimation are overwhelming [65, 25, 45, 44, 72]. The continuous-time model identification problem using digital computers, on the other hand, has yet to reach the same level although the relevance and importance of continuous-time system identification have been increasingly recognized in the recent years. An earlier survey solely devoted to this subject can be found in P. C. Young [85]. A comprehensive review of recent developments in the identification of continuous-time systems was given by Unbehauen and Rao [79]. A book written by Unbehauen and Rao [78] attempts to present a simple and unifying account of a broad class of identification techniques for continuous-time models.

Least squares (LS) is a basic technique for parameter estimation. The least squares principle, formulated by Gauss at the end of eighteenth century, says that the unknown parameters of a mathematical model should be chosen in such a way that

> *the sum of the squares of the differences between the actually observed and the computed values, multiplied by numbers that measure the degree of precision, is a minimum [64].*

The method is particularly simple if the model has the property of being *linear in the parameters.* That is if a model has the following form

$$y(t) = \varphi_1(t)\theta_1 + \varphi_2(t)\theta_2 + \cdots + \varphi_n(t)\theta_n = \varphi(t)^T\theta \tag{6.255}$$

where y is the observed variable, $\theta_1, \theta_2, \cdots, \theta_n$ are unknown parameters, and , $\varphi_1, \varphi_2, \cdots, \varphi_n$ are known functions that may depend on other known variables. The model is indexed by t, which often denotes time. The variables φ_i are called the regression variables or the regressors and the model described by (6.255) is called a regression model. The vectors $\varphi^T(t), \theta^T$ are defined as

$$\begin{aligned} \varphi^T(t) &= [\varphi_1(t) \quad \varphi_2(t) \quad \cdots \quad \varphi_n(t)] \\ \theta^T &= [\theta_1(t) \quad \theta_2(t) \quad \cdots \quad \theta_n(t)] \end{aligned} \tag{6.256}$$

With pairs of observations and regressors $\{(y(i), \varphi_i(i)),\ i = 1, 2, \cdots t\}$, the parameter estimation problem is to determine the parameters in such a way that the outputs computed from the model described by (6.255) agree as closely as possible with the measured variables $y(i)$. The solution has a analytical form

$$\hat{\theta} = \left(\Phi^T \Phi\right)^{-1} \Phi^T Y \tag{6.257}$$

where

$$\begin{aligned} Y(t) &= [y(1) \quad y(2) \quad \cdots \quad y(t)]^T \\ \Phi(t) &= \begin{bmatrix} \varphi^T(1) \\ \vdots \\ \varphi^T(t) \end{bmatrix} \end{aligned} \tag{6.258}$$

and the symbol "∧" denotes estimates throughout the thesis. In view of real time application, the computation based on (6.257) can also be done recursively resulting in the so called *Recursive Least Squares (RLS)* algorithm. The regression model (6.255) is an *algebraic* (non-dynamic) equation which has two important properties: it is realizable in the parameters θ^T and contains only realizable functions $\varphi^T(t)$ of the data. Usually there are two phases in applying the above least squares method to parameter estimation of dynamic models. The *primary phase* involves converting the *dynamic* equation into an *algebraic* equation. And the *secondary phase* involves solving these algebraic equations for the unknown parameters which is given by (6.257) or its recursive version. For systems described by a difference equation

$$\frac{y(t)}{u(t)} = \frac{b_0 q^n + b_1 q^{n-1} + \cdots + b_n}{a_0 q^n + a_1 q^{n-1} + \cdots + a_n} \tag{6.259}$$

or equivalently

$$\frac{y(t)}{u(t)} = \frac{b_0 + b_1 q^{-1} + \cdots + b_n q^{-n}}{a_0 + a_1 q^{-1} + \cdots + a_n q^{-n}} \tag{6.260}$$

where q is the difference operator, i.e. $qy(t) = y(t+1)$, the solution to the primary phase is obvious since the dynamic equation (6.260) already fulfills the requirement of the primary phase. For a continuous-time model given by

$$\frac{y(t)}{u(t)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_n}{a_0 s^n + a_1 s^{n-1} + \cdots + a_n} \tag{6.261}$$

where s is the differential operator, i.e. $sy(t) = dy(t)/dt$ (also loosely interpreted here as the Laplace operator), the solution to the primary phase is not as simple since derivative operation $s^n y(t) = d^n y(t)/d^n t$ is not feasible. One way to circumvent this difficulty is to make continuous-to-discrete conversion of the continuous-time model (6.261) first, and then estimate the parameters of the resulting discrete-time model. The parameters of the original continuous-time model can then be obtained by a discrete-to-continuous time transformation. However, obtaining a continuous-time model from its identified discrete-time form is not without difficulties [70, 71] as the choice of sampling interval is not trivial. On the other hand several methods are available to make the continuous-time model (6.261) compatible with the requirement of the primary phase without changing the parameters $(a_0 \cdots a_n \ , b_0 \cdots b_n)$ [79, 79, 85]. Perhaps the most direct of these is the low-pass filter approach. The key idea is to choose a low-pass filter $H(s)$ with a transfer function of sufficient relative degree to make $s^n H(s)$ proper so that all the signals $s^n y(t)$, $s^{n-1}y(t), \cdots, y(t)$, $s^n u(t)$, $s^{n-1}u(t), \cdots, u(t)$ involved in (6.261) will be feasible by passing through $H(s)$. Thus a *realizable*, linear-in-the-parameters form of equation (6.261) can be obtained. One particularly simple choice of the filter is the multiple integration filter $1/s^n$. The initial condition problem associated with integration operation can be overcome by integrating the input/output signal over a *moving* time interval $[t_0, t_0 + T]$ [67]. In section 6.1, the continuous-time model (6.261) is transformed into a regression model by passing $s^n y(t)$, $s^{n-1}y(t), \cdots, y(t)$, $s^n u(t)$, $s^{n-1}u(t), \cdots, u(t)$ through a multiple integration filter $1/s^n$. And the numerical integration formulae will also be given. Section 6.2 introduces the recursive least squares algorithm EFRA [68] which stands for *exponential forgetting and resetting algorithm.* In section 6.3, we develop a new algorithm to estimate fast time-varying parameters. A real life inverted pendulum experiment in section 6.4 shows the effectiveness of the algorithm presented in this chapter.

6.1 The Regression Model for Continuous-time Systems

Considering the system model (6.261), assuming the leading coefficient a_0 is equal to 1. Let $y^{(n)}(t)$ denotes the n-th derivative of $y(t)$, the system model being considered has the form

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y(t) = b_0 u^{(n)}(t) + b_1 u^{(n-1)}(t) + \cdots + b_n u(t) \qquad (6.262)$$

Define the multiple integral of signal $y(t)$ over $[t_0, t_0 + T]$ as

$$I_n y(t_0) = \int_{t_0}^{t_0+T} \int_{t_1}^{t_1+T} \cdots \int_{t_{j-1}}^{t_{j-1}+T} y(t_j) dt_j dt_{j-1} \cdots dt_1 \qquad (6.263)$$
$$j = 1, 2, \cdots n$$

Apply I_n defined in (6.263) to both sides of (6.262), the resulting regression model is

$$I_n y^{(n)}(t_0) = \varphi^T(t_0)\theta \qquad (6.264)$$

where

$$\begin{aligned} \varphi^T(t_0) &= [-I_n y^{(n-1)}(t_0) \quad \cdots \quad -I_n y(t_0) \quad I_n u^{(n)}(t_0) \quad \cdots \quad I_n u(t_0)] \\ \theta &= [\, a_1 \quad \cdots \quad a_n \quad b_0 \quad \cdots \quad b_n \,]^{\mathrm{T}} \end{aligned} \qquad (6.265)$$

As can be seen from (6.265), all the regressor entries are numerically feasible since derivation is no longer necessary. Since we are interested in computer implementation of the algorithm, the input/output data are only available at discrete sampling instants. However, this will be enough to compute the regressor numerically. The trapezoidal rule will be used for its simplicity in form and its satisfactory accuracy. Rather than giving the general formulae to compute the integrals in (6.265), we only consider the case when $n = 2$ for the purpose of not hindering the basic principle.

Assume $l + 1$ samples of signal $y(t)$, $y(0), y(1), \cdots y(l)$, are available on the interval $[t_0, t_0 + T]$ with sampling interval $\Delta = \frac{T}{l}$ as illustrated in Fig.6.43, then the integral $I_1 y(t_0) = \int_{t_0}^{t_0+T} y(t_1) dt_1$ can be given numerically using trapezoidal rule as

$$I_1y(t_0) = \int_{t_0}^{t_0+T} y(t_1)dt_1 \approx \Delta\left[\sum_{i=0}^{l} y(i) - 0.5(y(0)+y(l))\right] \tag{6.266}$$

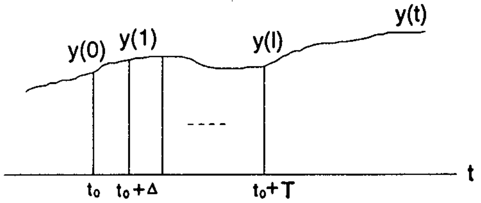

Figure 6.43: Graphical illustration of numerical integration

Similarly, the double integration of $y(t)$ over $[t_0, t_0+T]$ can be given by

$$\begin{aligned} I_2y(t_0) &= \int_{t_0}^{t_0+T} dt_1 \int_{t_1}^{t_1+T} y(t_2)dt_2 = \int_{t_0}^{t_0+T} (I_1y(t_1))dt_1 \\ &\approx \Delta\left[\sum_{i=0}^{l} I_1y(t_0+i\Delta) - 0.5I_1y(t_0) - 0.5I_1y(t_0+T)\right] \end{aligned} \tag{6.267}$$

where each $I_1y(t_0+i\Delta)$, $i = 0, 1, \cdots l$ in (6.267) can be computed using (6.266).

Double integration of $\ddot{y}(t)$ over $[t_0, t_0+T]$ is given by

$$\begin{aligned} I_2\ddot{y}(t_0) &= \int_{t_0}^{t_0+T} dt_1 \int_{t_1}^{t_1+T} \ddot{y}(t_2)dt_2 = \int_{t_0}^{t_0+T} (\dot{y}(t_1+T) - \dot{y}(t_1))dt_1 \\ &= y(t_0+2T) - 2y(t_0+T) + y(t_0) \end{aligned} \tag{6.268}$$

Double integration of $\dot{y}(t)$ over $[t_0, t_0+T]$ is given by

$$\begin{aligned} I_2\dot{y}(t_0) &= \int_{t_0}^{t_0+T} dt_1 \int_{t_1}^{t_1+T} \dot{y}(t_2)dt_2 = \int_{t_0}^{t_0+T} (y(t_1+T) - y(t_1))dt_1 \\ &= I_1y(t_0+T) - I_1y(t_0) \end{aligned} \tag{6.269}$$

With these formulae (6.266)-(6.269), the regressors in (6.265) can be computed for $n \leq 2$. For $n > 2$, the corresponding equations can be obtained in a similar fashion.

It is obvious that both the sampling interval Δ and the integration interval T affect the estimates. Sampling interval Δ directly affects the accuracy of the numerical integration (6.266) thus Δ should keep small. However, too small Δ may also lead to inaccurate estimates due to round off error. As a rule of thumb, the sampling interval should be chosen according to

$$\frac{T_s}{150} < \Delta < \frac{T_s}{10} \tag{6.270}$$

where $T_s = \frac{\pi}{w_n}$ is the Shannon maximum sampling period [86]. In obtaining the regression model (6.264), the multiple integral operation I_n defined in (6.263), which functions as a pre-filter, is applied to both sides of system model (6.262). Intuitively, the bandwidth of the this pre-filter should match that of the system (6.262) so that the noise in the measurement data can be depressed and at the same time, without rounding off the "richness" of the input/output signal which contains the information about the dynamics of the system (6.262). The Laplace transform of the multiple integrator (6.263) over a time length of T is [67]

$$\begin{aligned} \mathcal{L}[I_n(\cdot)] &= \frac{\left(1 - e^{-sT}\right)^n}{s^n} \mathcal{L}[(\cdot)] \\ &\triangleq \Xi(s)\mathcal{L}[(\cdot)] \end{aligned} \tag{6.271}$$

It is clear from (6.271) that the integration span T of the multiple integrator I_n should be selected such that the bandwidth of the multiple integrator transfer function $\Xi(s)$ matches the bandwidth of the system being identified.

6.2 The EFRA

For the regression model (6.264), the standard RLS algorithm is given by the following formulae[64]

$$\begin{aligned} \hat{\theta}(t_0) &= \hat{\theta}(t_0 - 1) + K(t_0)\Big(I_2 y^{(n)}(t_0) - \varphi^T(t_0)\hat{\theta}(t_0 - 1)\Big) \\ K(t_0) &= P(t_0)\varphi(t_0) = \frac{P(t_0 - 1)\varphi(t_0)}{I + \varphi^T(t_0)P(t_0 - 1)\varphi(t_0)} \\ P(t_0) &= \big(I - K(t_0)\varphi^T(t_0)\big)P(t_0 - 1) \end{aligned} \tag{6.272}$$

where $P(0)$ is a large enough positive definite matrix. The statistical interpretation of the least-squares method is that the initial covariance of the parameters is proportional to $P(0)$.

The RLS algorithm (6.272) can not track time-varying parameters effectively. A simple extension to (6.272) is to use a *forgetting factor* $0 < \lambda \leq 1$ to give more recent data more weight. The modified algorithm is given by

$$\begin{aligned}\hat{\theta}(t_0) &= \hat{\theta}(t_0-1) + K(t_0)\Big(I_2 y^{(n)}(t_0) - \varphi^T(t_0)\hat{\theta}(t_0-1)\Big)\\ K(t_0) = P(t_0)\varphi(t_0) &= \frac{P(t_0-1)\varphi(t_0)}{\lambda I + \varphi^T(t_0)P(t_0-1)\varphi(t_0)}\\ P(t_0) &= \big(I - K(t_0)\varphi^T(t_0)\big)P(t_0-1)/\lambda\end{aligned} \tag{6.273}$$

A disadvantage of the exponential forgetting (6.273) is that the covariance matrix P may eventually blow up when the excitation is poor. The Exponential Forgetting and Resetting Algorithm (EFRA) [68] of Salgado *et al* has been shown to have superior performance. It guarantees bounded covariance matrix P even when the excitation is poor.

The EFRA is given by

$$\begin{aligned}\hat{\theta}(t_0) &= \hat{\theta}(t_0-1) + K(t_0)\Big(I_2 y^{(n)}(t_0) - \varphi^T(t_0)\hat{\theta}(t_0-1)\Big)\\ K(t_0) &= \frac{\alpha P(t_0-1)\varphi(t_0)}{I + \varphi^T(t_0)P(t_0-1)\varphi(t_0)}\\ P(t_0) &= \frac{1}{\lambda}P(t_0-1) - K(t_0)\varphi^T(t_0)P(t_0-1) + \beta I - \delta P^2(t_0-1)\end{aligned} \tag{6.274}$$

There are four parameters in this algorithm to be chosen by the user. However, it is straightforward to select them in practice. The general guidelines are:

1. α adjusts the gain of the estimators, typically $\alpha \in [0.1, 0.5]$
2. β is small, directly related to the smallest eigenvalue of P, typically $\beta \in [0, 0.01]$
3. λ is the usual forgetting factor, $\lambda \in [0.9, 0.99]$
4. δ is small, inversely proportional to the maximum eigenvalue of P, typically $\delta \in [0, 0.01]$

The desirable features of EFRA are:

1. Exponential forgetting and resetting
2. An upper bound for P, i.e. a nonzero lower bound for P^{-1}
3. An lower bound for P^{-1}, i.e. a nonzero lower bound for P

6.3 Dealing with Fast Time-varying Parameters

RLS with forgetting factor can deal with slow time-varying parameters effectively but will encounter difficulties for fast time-varying parameters. In such cases it is advantageous to assume the parameters to be time-varying right from the start of the problem formulation. Xie and Evans [83] proposed an algorithm in a discrete-time setting assuming that the parameters are of the form of offset linear ramp. The moving horizon multiple integrator approach developed in section 6.1 will be used to deal with the time-varying parameter case. Again for simplicity, a second order differential equation with time varying parameters is considered.

Consider the equation

$$\ddot{y} + a_1(t)\dot{y} + a_2(t)y = b_1(t)u \tag{6.275}$$

Assume that the time-varying coefficients $a_i(t),\ b_i(t)$ have the form

$$\begin{aligned} a_i(t) &= a_{i0} + a_{i1}t \\ b_i(t) &= b_{i0} + b_{i1}t \\ t &\in [0, T_{res}] \end{aligned} \tag{6.276}$$

over interval $[t_0,\ \ t_0 + T_{res}]$. Obviously, equation (6.276) would be a very good approximation of $a_i(t),\ b_i(t)$ if T_{res} is reasonably small. Note that T_{res} is not necessarily the same as the integration span T in (6.263). Usually $T_{res} > T$.

Substitute equation (6.276) into equation (6.275), we have

$$\ddot{y} + a_{10}\dot{y} + a_{20}y + a_{11}t\dot{y} + a_{21}ty = b_{10}u + b_{11}tu \tag{6.277}$$

Apply I_2 defined in (6.263) on both sides of (6.277) over $[t_0,\ \ t_0 + T]$, the following regression model can be obtained

$$I_2\ddot{y}(t_0) = \varphi^T(t_0)\theta \tag{6.278}$$

where

$$\begin{aligned}\varphi^T(t_0) &= \left[-I_2\dot{y}(t_0) \quad -I_2y(t_0) \quad I_2u(t_0) \; \vdots \; -I_2t\dot{y}(t_0) \quad -I_2ty(t_0) \, \vdots \, I_2tu(t_0)\right] \\ \theta &= \left[a_{10} \quad a_{20} \quad b_{10} \, \vdots \, a_{11} \quad a_{21} \quad b_{11}\right]^{\mathrm{T}}\end{aligned} \tag{6.279}$$

The integrals $I_2\ddot{y}(t_0), I_2\dot{y}(t_0), I_2y(t_0), I_2u(t_0)$ can be computed using formulae (6.266)-(6.269), while $I_2t\dot{y}(t_0), I_2ty(t_0), I_2tu(t_0)$ are given as follows

$$\begin{aligned}I_2t\dot{y}(t_0) &= \int\limits_{t_0}^{t_0+T} dt_1 \int\limits_{t_1}^{t_1+T} \tau\dot{y}(\tau)d\tau = \int\limits_{t_0}^{t_0+T} dt_1 \int\limits_{t_1}^{t_1+T} \tau dy(\tau) \\ &= \int\limits_{t_0}^{t_0+T} \left[(\tau y(\tau))|_{t_1}^{t_1+T} - \int\limits_{t_1}^{t_1+T} y(\tau)d\tau\right] dt_1 \\ &= \int\limits_{t_0}^{t_0+T} [(t_1+T)y(t_1+T) - t_1y(t_1) - I_1y(t_1)]dt_1 \\ I_2ty(t_0) &= \int\limits_{t_0}^{t_0+T} dt_1 \int\limits_{t_1}^{t_1+T} \tau y(\tau)d\tau \\ I_2tu(t_0) &= \int\limits_{t_0}^{t_0+T} dt_1 \int\limits_{t_1}^{t_1+T} \tau u(\tau)d\tau\end{aligned} \tag{6.280}$$

With the regression model (6.278), either the standard RLS (6.272) or the EFRA (6.274) can be used.

Recall equation (6.276), apparently the offset linear ramp approximation of time-varying coefficients is valid only when T_{res} is small. Define T_{res} as the *resetting period* . At each kT_{res}, $k = 0, 1, 2\cdots$, it is necessary to reset parameter vector and covariance matrix as follows

$$\begin{aligned}
a_{10}(k+1) &= a_{10}(k) + T_{res}a_{11}(k) \\
a_{11}(k+1) &= a_{11}(k) \\
a_{20}(k+1) &= a_{20}(k) + T_{res}a_{21}(k) \\
a_{21}(k+1) &= a_{21}(k) \\
b_{10}(k+1) &= b_{10}(k) + T_{res}b_{11}(k) \\
b_{11}(k+1) &= b_{11}(k) \\
P(k+1) &= K_1 P(k), \quad K_1 > 1
\end{aligned} \tag{6.281}$$

It is important to select the resetting period T_{res} properly, the principle is that T_{res} must be chosen large enough to allow reasonable convergence of the parameters but the variation of the real parameters over the period of T_{res} should stay small so that the offset linear ramp is still a good approximation. The following example shows the effectiveness of this algorithm.

Example 6.3.1

Consider system (6.275) with parameters described by

$$\begin{aligned}
b(t) &= 2 + 0.1t, \quad 0 \le t \le 30sec \\
a_1(t) &= 2 + 1.5 * sin(0.2\pi t), \quad 0 \le t \le 30sec \\
a_2 &= 1, \quad 0 \le t \le 30sec
\end{aligned} \tag{6.282}$$

The simulation is performed in open-loop with PRBS signal as input, and the following settings: sampling interval $\Delta = 0.01sec$, integration interval $T = 0.05sec$ and resetting period $T_{res} = 0.08sec$. The standard RLS algorithm (6.272) is used.

The results are depicted in Fig.6.44.

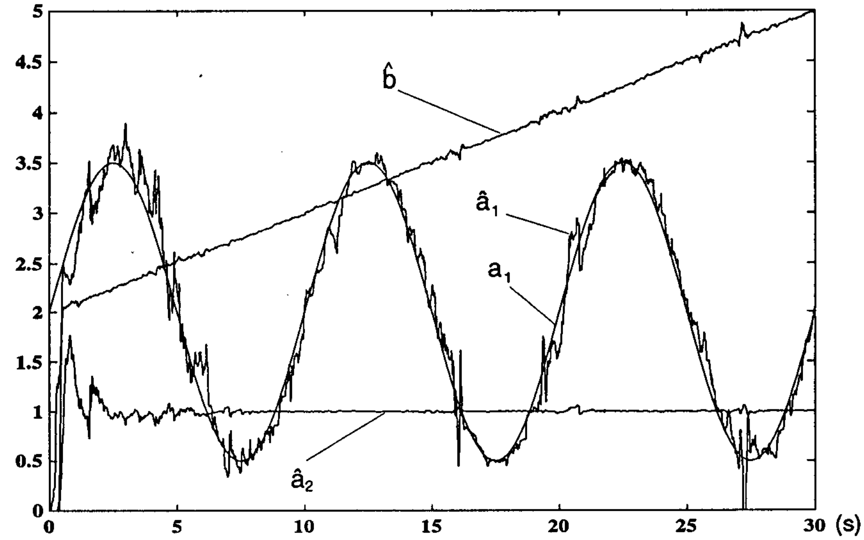

Figure 6.44: Estimation of time-varying parameters

As can be seen from Fig.6.44, even the sinusoidally time-varying parameter can be tracked satisfactorily. This verifies the validity of our assumption in equation (6.276).

6.4 Identification and Control of an Inverted Pendulum

The control of an inverted pendulum is a classic topic in control engineering. There are many solutions available to this problem, for example *PID*, *LQG*, fuzzy logic etc. The SDGPC solution will be given in this section. The advantage of continuous-time parameter estimation over the discrete-time method is highlighted by the comparison between the two different approaches.

Fig 6.45 is a picture of the experimental setup[2] which is built on a used printer. The pendulum rod is mounted on the printer head and able to freewheel 360 degrees around its axis. The printer head is driven by a DC motor along the x axis. Both the printer head position x and the pendulum angel θ are available for measurement through a LVDT and an encoder attached to the printer head. The control input to the system is the voltage applied to the DC motor. The purpose of the control is to keep the pendulum rod upward and at the same time keep the printer head at the center position.

[2] The author thanks Dr. A. Elnaggar who was then a research engineer at the Pulp & Paper Centre for making this experimental setup available.

Figure 6.45: The inverted pendulum experimental setup

6.4.1 System Model

The printer head position is proportional to the angular displacement of the DC motor. Thus the transfer function from the input voltage $u(t)$ to the printer head position $x(t)$ has the form

$$G_m(s) = \frac{x(s)}{u(s)} = \frac{k_m}{s(\tau_m s + 1)} = \frac{b_0}{s^2 + a_1 s} \tag{6.283}$$

Only two parameters b_0, a_1 need to be estimated.

As for the relation between the printer head position $x(t)$ and the pendulum angle $\theta(t)$, let us consider the downward pendulum first.

Fig. 6.46 is an idealized sketch of the downward pendulum. Notice that at the equilibrium point $\theta_0 = 0$, only the *acceleration* of the printer head M will break the equilibrium. Suppose M is moving with acceleration $\ddot{x}(t)$, observing on the moving M, the effect of $\ddot{x}(t)$ on m is that it seems as though there is a force $m\ddot{x}(t)$ being applied on the other direction. Applying Newton's law in the θ direction as indicated in Fig. 6.46, the torque balance is

$$-mg * L\sin\theta - \varepsilon\dot{\theta} + m\ddot{x}(t) * L\cos\theta = mL^2\ddot{\theta} \tag{6.284}$$

where L, m are the length and the mass of the idealized pendulum respectively. g is the gravity constant and ε is the friction coefficient. Assuming small θ, and linearizing it around $\theta_0 = 0$, equation (6.284) becomes

$$\begin{aligned} \ddot{\theta} + a_1\dot{\theta} + a_2\theta &= b\ddot{x}(t) \\ G_{down}(s) = \frac{\theta(s)}{x(s)} &= \frac{bs^2}{s^2 + a_1 s + a_2} \\ a_1 = \frac{\varepsilon}{mL^2},\ a_2 &= \frac{g}{L},\ b = \frac{1}{L} \end{aligned} \tag{6.285}$$

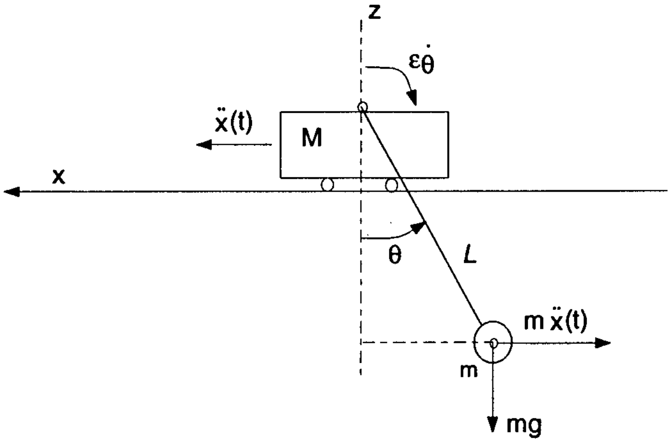

Figure 6.46: Downward pendulum

Similarly, the torque balance for the upward pendulum case is

$$mg * L\sin\theta - \varepsilon\dot{\theta} + m\ddot{x}(t) * L\cos\theta = mL^2\ddot{\theta} \tag{6.286}$$

And the linearized model around $\theta_0 = 0$ can be written as

$$
\begin{aligned}
&\ddot{\theta} + a_1\dot{\theta} - a_2\theta = b\ddot{x}(t) \\
&G_{up}(s) = \frac{\theta(s)}{x(s)} = \frac{bs^2}{s^2 + a_1 s - a_2} \\
&a_1 = \frac{\varepsilon}{mL^2},\ a_2 = \frac{g}{L},\ b = \frac{1}{L}
\end{aligned}
\tag{6.287}
$$

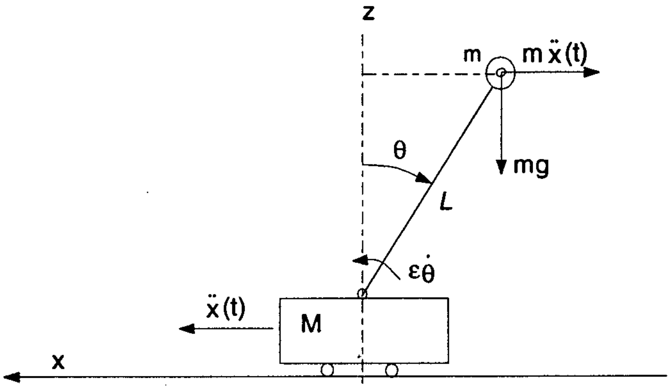

Figure 6.47: Upward pendulum

Comparing $G_{down}(s)$ in (6.285) and $G_{up}(s)$ in (6.287), it is easy to see that the parameters in . $G_{down}(s)$ and $G_{up}(s)$ are the same except for a sign difference in a_2. This important *a priori* information is only preserved in the continuous-time model! Since the upward pendulum is open loop unstable, it is very difficult, if not impossible, to estimate $G_{up}(s)$ directly without stabilizing it first. However, with continuous-time modeling, it is possible to estimate a_1, a_2 and b for the downward pendulum which is open-loop stable. The estimation results will be presented in the next subsection.

6.4.2 Parameter Estimation

The experiment was conducted on the downward pendulum. The input voltage applied to the DC motor is PRBS (Pseudo Random Binary Sequence) signal with an amplitude of ± 1 volt and a length of $2^8 - 1 = 255$ samples. The sampling interval $\Delta = 0.1sec$. Both the printer head position $x(t)$ and the angle $\theta(t)$ are measured. For the model structure given by (6.283), the continuous-time regression model (6.264) and the standard RLS (6.272) can be readily applied. The integration interval $T = 0.3sec$. The input/output data and the estimated parameters $\hat{b}_0, \hat{a}_1$ are depicted in Fig. 6.48 where $\hat{b}_0 = 10250, \hat{a}_1 = 12.59$.

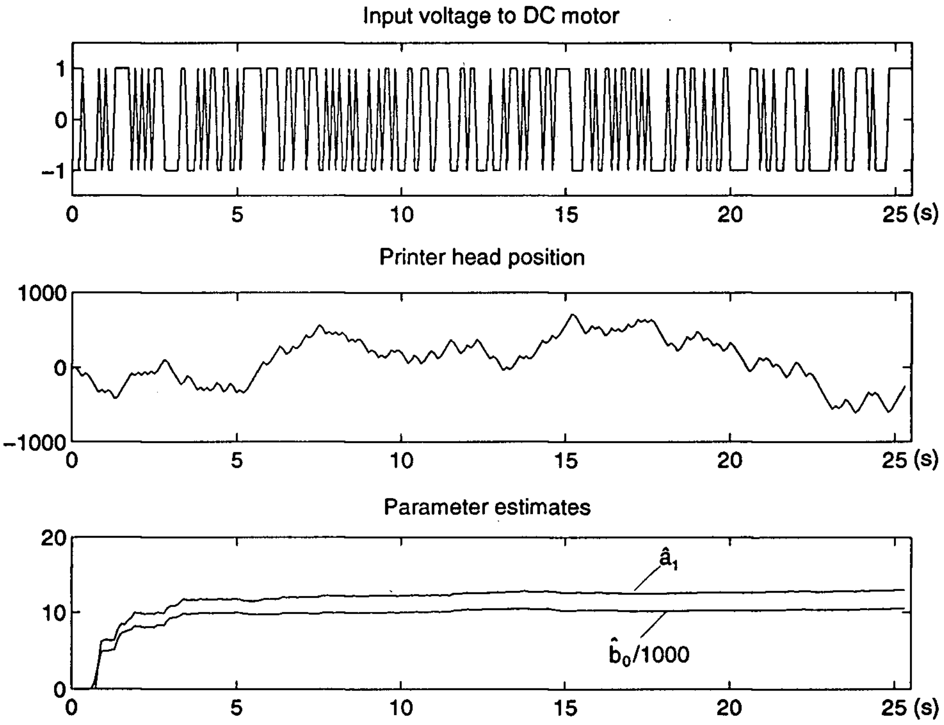

Figure 6.48: Parameter estimation of model (6.283)

The estimated model from input voltage to printer head position is thus

$$\hat{G}_m = \frac{10250}{s^2 + 12.59s} = \frac{814.09}{s(0.0794s + 1)} \tag{6.288}$$

Since the time constant $\tau_m = 0{:}0794sec$ is very small, (6.288) can be reasonably approximated by

$$\hat{G}_m = \frac{814.09}{s} \tag{6.289}$$

For the identification of the system model from printer head position to pendulum angle, a 255 sample PRBS signal with an amplitude of $1v$ is applied to the motor. The sampling interval is again $\Delta = 0.1sec$. The model parameters a_1, a_2 and b of $G_{down}(s)$ in (6.285) are estimated using algorithms given in previous sections in this chapter. The printer head position, the angle output data and the estimated parameters $\hat{a}_1, \hat{a}_2, \hat{b}$ are depicted in Fig. 6.49 with $\hat{a}_1 = 0.01418, \hat{a}_2 = 43.9627, \hat{b} = 0.1125$.

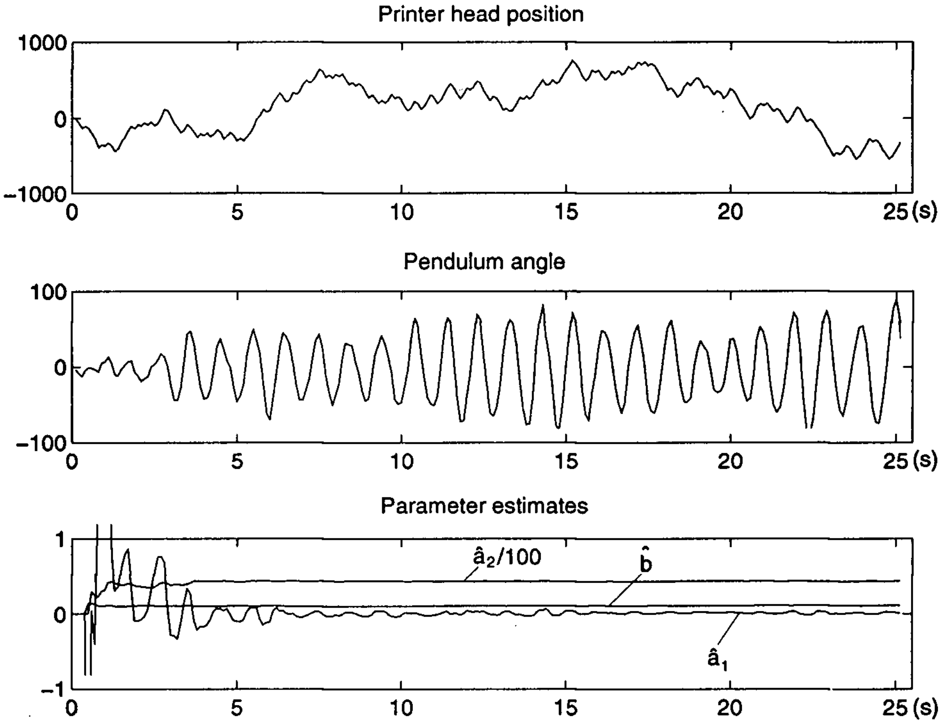

Figure 6.49: Parameter estimation of model (6.283)

The identified model $\hat{G}_{down}(s)$ is thus

$$\hat{G}_{down}(s) = \frac{0.1125s^2}{s^2 + 0.01418s + 43.9627} \tag{6.290}$$

with poles at $-0.0071 \pm 6.6304i$.

The open loop unstable upward pendulum model (6.287) can be readily written as

$$\hat{G}_{up}(s) = \frac{0.1125s^2}{s^2 + 0.01418s - 43.9627} \tag{6.291}$$

with poles at $-6.6375, 6.6233$ which correspond to a time constant of approximately $0.15sec$. The sampling interval of $\Delta = 0.1sec$ we used in the experiment is relatively large for this system, either for the downward or the upward case. Unfortunately, that was the smallest sampling interval we can get due to the computer system limitations.

It is interesting to see how discrete-time estimation will perform with the same experimental data. The discrete-time counterpart of system model (6.285) has the form:

$$\begin{aligned} &\theta(k) + a_{d_1}\theta(k-1) + a_{d_2}\theta(k-2) = \\ &b_{d_1}x(k) + b_{d_2}x(k-1) + b_{d_3}x(k-2) \\ G_{q_d}(q) = &\frac{\theta(k)}{x(k)} = \frac{b_{d_1} + b_{d_2}q^{-1} + b_{d_3}q^{-2}}{1 + a_{d_1}q^{-1} + a_{d_2}q^{-2}} \end{aligned} \tag{6.292}$$

The *MATLAB* function *ARX* in the system identification toolbox was used on the same data set to identify the parameters in (6.292), i.e.:

$$\begin{gathered} arx([\theta \quad x],[n_a \quad n_b \quad n_k]) \\ n_a = 2, n_b = 3, n_k = 0 \end{gathered} \tag{6.293}$$

The parameter estimates are

$$\begin{gathered} [\hat{b}_{d_1} \quad \hat{b}_{d_2} \quad \hat{b}_{d_3}] = [0.093 \quad -0.1863 \quad 0.0923] \\ [\hat{a}_{d_1} \quad \hat{a}_{d_2}] = [-1.5685 \quad 0.9663] \end{gathered} \tag{6.294}$$

Fig. 6.50 shows the step response of model (6.292) with estimated parameters.

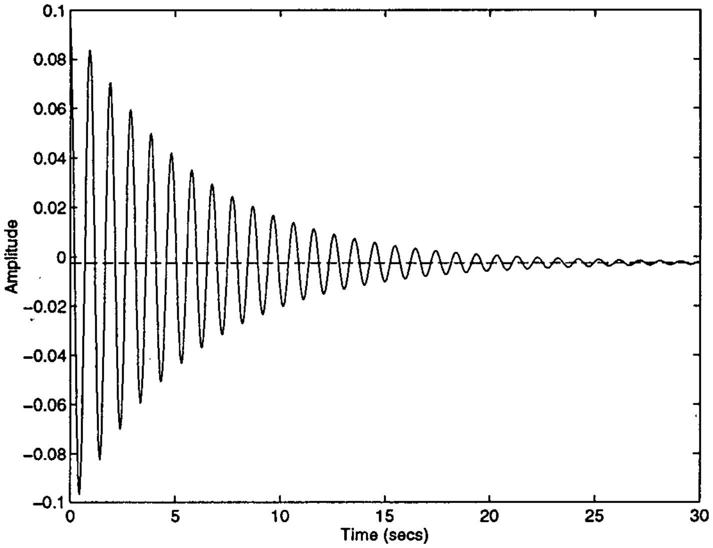

Figure 6.50: Step response of the estimated discrete-time model (6.292)

Fig. 6.50 tells us that the system has a natural undamped frequency of 1.05. This agrees quite satisfactorily with what we have from the continuous-time identification approach. See (6.290) where

$f = \frac{\sqrt{43.9627}}{2\pi} = 1.0553$. However, the damping factor is quite different from what we obtained in (6.290). From Fig. 6.50, the pendulum should settle in about 30 seconds when there is a step position input. Experiment shows that the pendulum oscillation will last about 400 seconds after a hit on the printer head which agrees quite well with the continuous-time estimation results. Also notice from Fig. 6.50 that there is a nonzero steady state gain in the estimated discrete-time model. This is obviously wrong.

It has been shown that the system has two zeros at the origin, see (6.285). This important *a priori* information is also lost in the discrete-time modelling. As a matter of fact, the continuous-time counterpart of the estimated discrete-time model with sampling interval $\Delta = 0.1sec$ and zero-order-hold is:

$$\frac{0.093s^2 + 0.2004s - 0.1036}{s^2 + 0.3428s + 41.91} \tag{6.295}$$

Compare (6.295) with (6.290), the superior performance of continuous-time identification for this example is obvious.

6.4.3 Controller Design

Define the states of the pendulum system and the input voltage to the DC motor as $[\dot{\theta}(t) \quad \theta(t) \quad \dot{x}(t) \quad x(t)]^T$ and $u(t)$ respectively, the state space description of the system can be written based on (6.289) and (6.291)

$$\begin{bmatrix} \dot{\theta} \\ \theta \\ \dot{x} \\ x \end{bmatrix} = \begin{bmatrix} -0.01418 & 43.9627 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \theta \\ \dot{x} \\ x \end{bmatrix} + \begin{bmatrix} 91.5851 \\ 0 \\ 814.09 \\ 0 \end{bmatrix} u_d(t) \tag{6.296}$$

where $u_d(t) = \dot{u}(t)$.

In the SDGPC framework, system (6.296) can be regarded as the integrator augmented system of

$$\begin{bmatrix} \dot{\theta} \\ \overset{t}{\int} \theta \\ x \end{bmatrix} = \begin{bmatrix} -0.01418 & 43.9627 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \overset{t}{\int} \theta \\ x \end{bmatrix} + \begin{bmatrix} 91.5851 \\ 0 \\ 814.09 \end{bmatrix} u(t)$$

$$x(t) = [0 \quad 0 \quad 1] \begin{bmatrix} \theta \\ \overset{t}{\int} \theta \\ x \end{bmatrix} \tag{6.297}$$

Apply the final state weighting SDGPC (2.20) to (6.296) with design parameters:

$$\begin{aligned} N_u &= 6 \\ T_p &= 1.2 sec \\ \lambda &= 1 \\ \gamma &= 1 \end{aligned} \tag{6.298}$$

The resulting control law is

$$u_d(t) = -[0.1905 \quad 1.2643 \quad -0.0069 \quad -0.0120] \begin{bmatrix} \dot{\theta} \\ \theta \\ \dot{x} \\ x \end{bmatrix} \tag{6.299}$$

The control law (6.299) is implemented in the following form:

$$\begin{aligned} u(t) &= -0.1905 * \theta(t) - 1.2643 * \int^{t} \theta(\tau) d\tau \\ &+ 0.0069 * (x(t) - 1100) + 0.0120 * \int^{t} (x(\tau) - 1100) d\tau \end{aligned} \tag{6.300}$$

The number 1100 in (6.300) is the LVDT reading corresponding to the center position.

The picture in Fig.6.52 shows the pendulum being successfully controlled by (6.300). Notice that the control law is fairly robust against a disturbance (a plastic bag was placed on the top of the pendulum after (6.300) was applied). The same control law can also stabilize the pendulum when the rod is stretched to twice the original length. See Fig. 6.51.

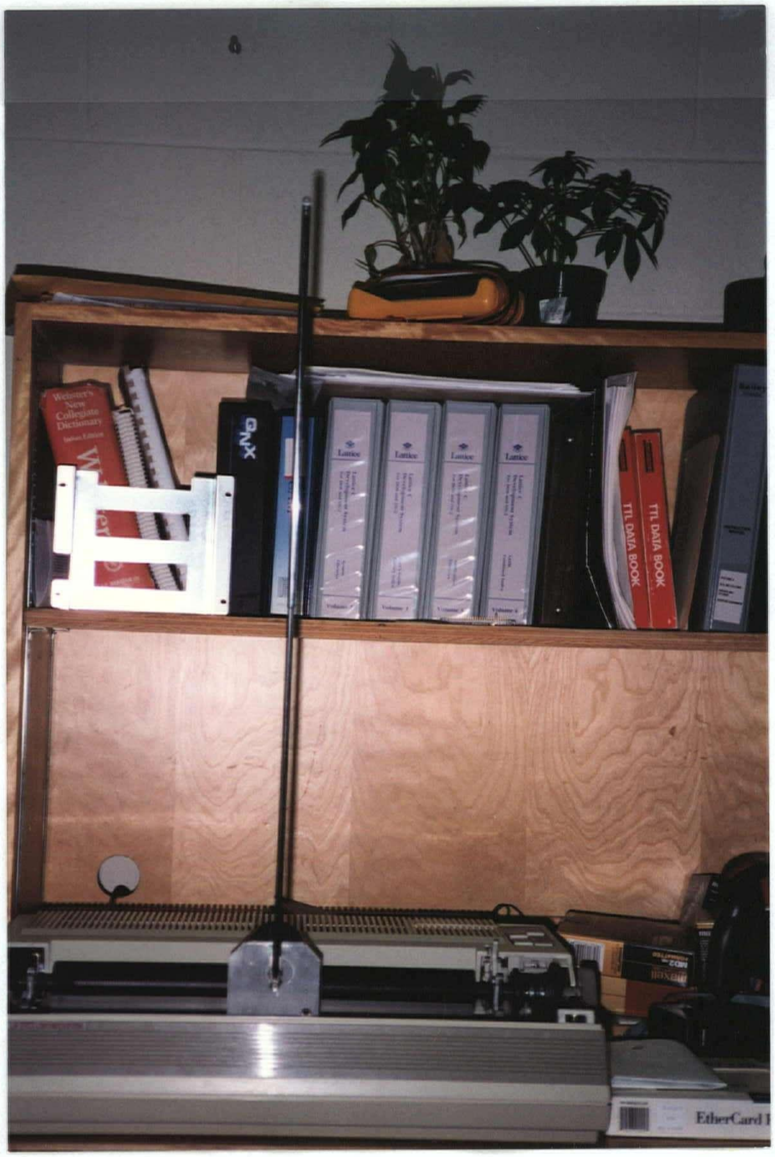

Figure 6.51: Changing dynamics

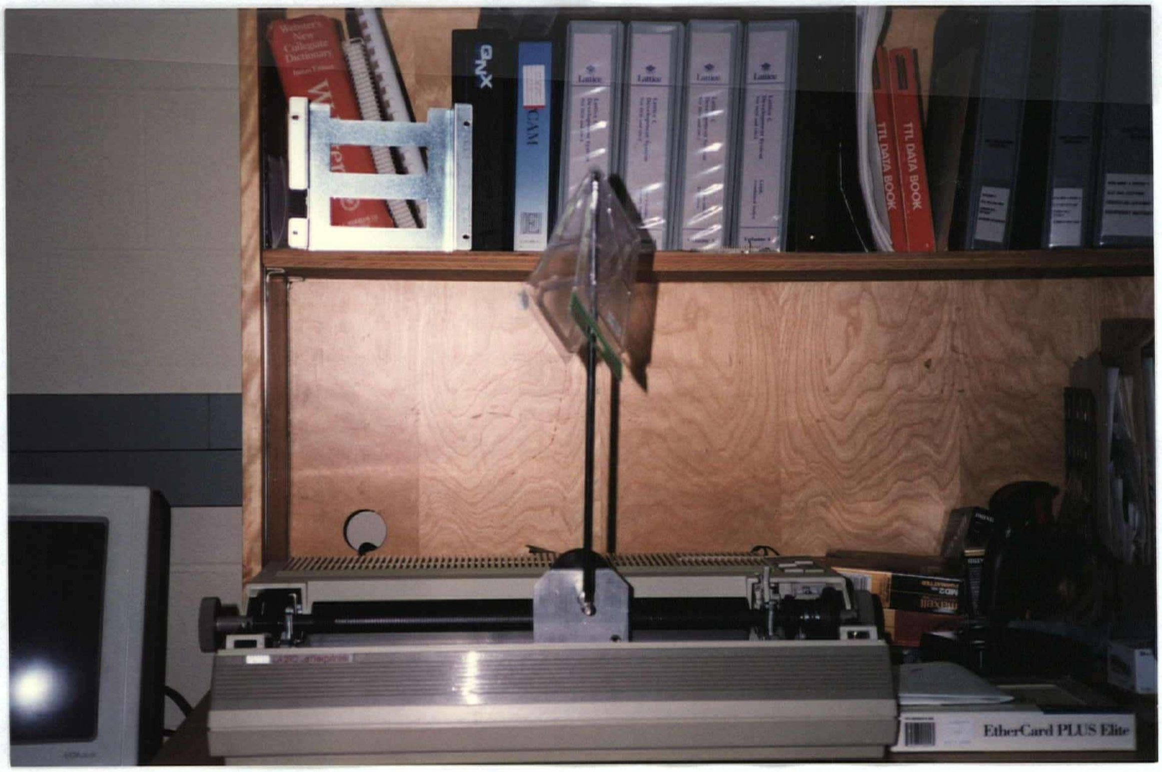

Figure 6.52: SDGPC of pendulum subject to disturbance

It is found in the experiment that it is not difficult to stabilize the pendulum, i.e. keeping it upward. But it is not easy to control the printer head exactly in the center position while keeping the pendulum upward. Also notice that the system model we developed in subsection 6.4.1 is an idealized one with the assumption that the pendulum rod is a rigid body. As the length of the rod increases, so does its flexibility. An adaptive version of the SDGPC would have been more interesting. Unfortunately the experiment setup was only available for a limited period of time. Nevertheless, the experiment results show that the continuous-time model parameter estimation algorithm and the SDGPC algorithm are quite effective solving practical control problems.

6.5 Conclusion

Continuous-time system identification based on sampled-data is considered in this chapter. The moving horizon integration approach given in section 6.1 is a simple yet powerful method for parameter estimation of a continuous-time model. Based on the regression model (6.264), various available recursive estimation algorithms such as the EFRA developed in discrete-time context can be readily applied. The algorithm we proposed in 6.2 can deal with fast time-varying parameters as was shown by simulation. A real life inverted pendulum experiment in section 6.3 showed the benefits of continuous-time identification, namely, effect use of *a priori* information etc. The identification method in this chapter together with the SDGPC algorithm offer an effective way for solving complicated control problems. In this way, the insight about the underlying inherent continuous-time process is never lost during the whole design process. It is the author's belief that even if the control law is designed in discrete-time domain, it is always beneficial to identify the underlying continuous-time process first and then discretize it. The experiment presented in section 6.3 can be regarded as a supportive example.

Chapter 7

Conclusions

A new predictive control approach was taken in this thesis. The important issue such as actuator saturation in practical applications was taken into account. The resulting algorithms have very important practical interests as well as nice theoretical properties. The work can be summarized as follows.

1. A new predictive control algorithm, SDGPC, has been developed. It possesses the inherent robustness (gain and phase margin) and stability property of infinite horizon LQ regulator and at the same time, has the constraint handling flexibility of the finite horizon formulation, a feature unique to MBPC. SDGPC distinguishes itself from the rest of the MBPC family in that it is based on continuous-time modelling yet assumes digital implementation. This formulation stresses the connection rather than the differences between continuous-time and discrete-time control. It has been shown by simulation that for a stable well damped process, the execution time T_{exe} can vary from 0, which corresponds to continuous-time control, to the design sampling interval T_m, which can be quite large, without affecting the servo performance significantly. This means that for a given prediction horizon T_p and desired sampling interval T_{exe}, a larger T_m can be selected to reduce computation burden in adaptive applications. For unstable and/or lightly damped processes, however, T_{exe} should be equal to T_m.
2. SDGPC for tracking systems has a two-degree-of-freedom design structure. This is achieved by assuming a different model for the reference signal and the disturbance. However, only one performance index is used to obtain the control law. Tracking performance can be improved radically when the future setpoint information is available. This is because knowing the future setpoint is equivalent to knowing the complete state information of the reference signal at present time.
3. Another two-degree-of-freedom design extension to SDGPC was made. Contrary to the approach taken in tracking system design in which different models for reference and disturbance were assumed, two performance indices are used but assuming the reference and the disturbance

have the same model (in this thesis, it is a simple constant). The servo performance and the disturbance rejection performance can be tuned separately by using different design parameters (prediction horizon, control order, control weighting etc.) for the two performance indices. The nonlinearity due to actuator constraints is considered in the framework of anti-windup design. The resulting scheme effectively transforms the nonlinear problem into a time-varying linear problem and was shown to have guaranteed stability property. Simulation results confirmed the effectiveness of the scheme.

4. Control of time-delay systems was considered. Laguerre filter based adaptive SDGPC was shown to be particularly effective in dealing with time-delay systems. *A priori* information about the time-delay can be utilized to improve the control performance significantly.
5. A continuous-time model parameter estimation algorithm based on sampled data was developed. Numerical integration on a moving interval was used to eliminate the initial condition problem. It was argued that even if the controller design is purely in discrete-time, it is always beneficial to identify the underlying continuous-time model first before discretizing. *A priori* information about the physical system is best utilized in continuous-time modelling. The continuous-time model identification method and the SDGPC algorithm were applied to an inverted pendulum experiment. The results confirmed the benefits of continuous-time modelling and identification.

Some future research suggestions are:

1. Extend the work to multi-input multi-output systems. Although the author sees no major obstacles in doing so for most of the topics covered in the thesis, some efforts are needed to formulate the anti-windup scheme for MIMO case.
2. Dealing with the trade-off between good tracking and disturbance rejection performance and, good noise suppression performance. This is a basic trade-off in any control systems design [3, pp. 112]. SDGPC was formulated in a deterministic framework. Deterministic disturbances such as impulse, step, ramp, sinusoidal etc. can all be handled in a straightforward manner in the framework of SDGPC. The trade-off between good noise suppress performance and good disturbance rejection performance can be obtained by proper tuning of SDGPC to give the desired

closed-loop system bandwidth. For stochastic noise, the well known *Kalman filter* theory can be applied to estimate the system states. The deterministic treatment of SDGPC does not prevent it from using these results because of the *Separation Theorem or Certainty Equivalence Principle* [3, pp. 218]. For systems with colored noise, which is more often than not, the optimal *Kalman filter* can be designed based on the noise model provided that the noise model is known [4, pp. 54]. Unfortunately, the noise model is often unknown and difficult to estimate. Estimation of the "true" system states subject to unknown colored noise poses one of the biggest challenges in process control applications. Thus how to utilize the available results and develop new one, in the framework of SDGPC, is certainly a topic worth pursuing.

3. Adaptive SDGPC. We only considered adaptive Laguerre filter based SDGPC in the thesis. Since the parameter estimation algorithm has been developed, it would be nice to see an adaptive version of SDGPC based on general transfer function description of systems.
4. Apply the SDGPC algorithm to practical problems. Although initial experiment on an inverted pendulum showed the effectiveness of SDGPC and the associated continuous-time identification algorithm, only large scale industrial applications can be the final judge.

Appendix A Stability Results of Receding Horizon Control

Appendix A.1 and appendix A.2 are based on [4].

A.1 The Finite and Infinite Horizon Regulator

Given a state-space model of a linear plant where F, G, H have proper dimensions.

$$x(k+1) = Fx(k) + Gu(k) \tag{A.301a}$$

$$y(k) = Hx(k) \tag{A.301b}$$

The finite horizon LQ regulator problem can be posed as follows.

The performance index:

$$J(N, x(k)) = x^T(k+N)P_0x(k+N) + \sum_{j=0}^{N-1} \left\{ x^T(k+j)Qx^T(k+j) + u^T(k+j)Ru(k+j) \right\} \tag{A.302}$$

The solution to the above optimal LQ problem may be given by iterating the Riccati Difference Equation (RDE),

$$\begin{gathered} P_{j+1} = F^TP_jF - F^TP_jG\left(G^TP_jG + R\right)^{-1}G^TP_jF + Q \\ j = 0, 1, \cdots N-2 \end{gathered} \tag{A.303}$$

from the initial condition P_0 and implements the feedback control sequences given by

$$\begin{aligned} u(k+N-j) &= -\left(G^TP_{j-1}G + R\right)^{-1}G^TP_{j-1}Fx(k+N-j) \\ &= K_{j-1}x(k+N-j), \ \ j = 1, 2, \cdots N, \end{aligned} \tag{A.304}$$

where P_j is the matrix solution of RDE (A.303). Notice from the control sequence (A.304) that it iterates reversely in time compared with the direction of evolution of the plant (A.301). That is, in order to obtain the current control $u(k)$, P_{N-1} has to be solved first by iterating (A.303). The accumulated cost of (A.302) is given by P_N which itself does not appear in the control law,

$$J(N, x(k))^F = x^T(k) P_N x(k) \tag{A.305}$$

Similarly, the infinite horizon LQ regulator problem may be posed as the limiting case of the finite horizon LQ problem (A.302),

$$J(x(k)) = \lim_{N \to \infty} J(N, x(k)) \tag{A.306}$$

And the optimal solution can be obtained by iterating (A.303) indefinitely. Under mild assumptions, P_j converges to its limit P_∞ which is the maximal solution of the Algebraic Riccati Equation (ARE),

$$P_\infty = F^T P_\infty F - F^T P_\infty G \left(G^T P_\infty G + R\right)^{-1} G^T P_\infty F + Q \tag{A.307}$$

And a stationary control law is obtained as

$$u(k) = -\left(G^T P_\infty G + R\right)^{-1} G^T P_\infty F x(k) = K x(k) \tag{A.308}$$

The following theorem regarding the stability property of the infinite horizon LQ control law (A.308) is due to De Souza *et al* [74].

Theorem **A.6** (De Souza *et al* [74])

Consider an infinite horizon LQ regulator problem with plant (A.301) and performance index (A.306), for the associated ARE,

$$P = F^T PF - F^T PG\left(G^T PG + R\right)^{-1} G^T PF + Q \tag{A.309}$$

if

- $[F, G]$ *is stabilizable,*
- $\left[F, Q^{1/2}\right]$ *has no unobservable modes on the unit circle,*
- $Q \geq 0$ *and* $R > 0$,

then

- *there exists a unique, maximal, nonnegative definite symmetric solution* $\bar{P}$.
- $\bar{P}$ *is a unique stabilizing solution, i.e.*

$$F - G\left(G^T \bar{P} G + R\right)^{-1} G^T \bar{P} F \tag{A.310}$$

has all its eigenvalues strictly within the unit circle.

The solution $\bar{P}$ above is called the stabilizing solution of the ARE (A.309). Also note that the matrix (A.310) is the state transition matrix of the closed loop system when the stationary control law (A.308) is applied to plant (A.301). ***Theorem*** A.6 is the fundamental closed loop stability result for infinite horizon LQ control which will be utilized to prove the stability result of receding horizon control in the following and the stability property of SDGPC thereafter.

A.2 The Receding Horizon Regulator

From the discussions in appendix A.1, a number of facts about the finite horizon and infinite horizon discrete-time LQ regulator problem are clear. For the finite horizon case, the optimization task with cost function (A.302) is merely to find N control values which, in principle, may be found by finite-dimensional optimization which is referred as the "one shot" algorithm in most model based predictive controllers. The control sequences may also be obtained by iterating the RDE (A.303) explicitly from P_0 to P_{N-2} using simple linear algebra. The resulting control law in feedback form (A.304) is time-varying even if the plant being controlled is time invariant. By contrast, the infinite horizon problem involves an infinite-dimensional optimization or the solution of an ARE (A.307) which is computationally burdensome especially in adaptive applications. However, the control law of the infinite horizon problem is stationary and have guaranteed stability properties under mild assumptions.

Receding horizon control is one method proposed to inherit the simplicity of the finite horizon LQ method while addressing an infinite horizon implementation and preserving the time-invariance of the infinite horizon feedback. In this formulation only the first element $u(k)$ in the control sequences $u(k), u(k+1), \cdots u(k+N-1)$ is applied to the plant at time k and at time $k+1$ the first control $u(k+1)$ in the control sequences $u(k+1), u(k+2), \cdots u(k+N)$ is applied and so on. In terms of the finite horizon feedback control law (A.304), one has for the receding horizon strategy

$$\begin{aligned} u(k) &= K_{N-1}x(k) \\ &= -\left(G^T P_{N-1} G + R\right)^{-1} G^T P_{N-1} F x(k) \end{aligned} \tag{A.311}$$

which is a stationary control law.

Note that there is still no word having been said about the stability of control law (A.304). In fact, receding horizon strategy does not guarantee stability itself. Motivated by the facts that the infinite horizon LQ control law has guaranteed stabilizing property and there are strong similarities between receding horizon control law (A.304) and infinite horizon LQ control law (A.308), i.e. both are stationary and have the same form, one has enough reason to wonder if the stability result of infinite horizon LQ control summarized as ***Theorem*** A.6 could be of any help to the stability problem of receding horizon control. For this we go to the important work of Bitmead *et al* [4].

Consider the RDE (A.303)

$$\begin{gathered} P_{j+1} = F^T P_j F - F^T P_j G\left(G^T P_j G + R\right)^{-1} G^T P_j F + Q \\ j = 0, 1, \cdots N-2 \end{gathered} \tag{A.312}$$

define

$$\bar{Q}_j = Q - (P_{j+1} - P_j) \tag{A.313}$$

, the RDE (A.312) will have a form of an ARE,

$$P_j = F^T P_j F - F^T P_j G\left(G^T P_j G + R\right)^{-1} G^T P_j F + \bar{Q}_j \tag{A.314}$$

which is called Fake Algebraic Riccati Equation (FARE) [4].

From ***Theorem*** A.6, the stability property of the solution of the above FARE can be immediately established as follows.

Theorem **A.7** (Bitmead *et al.* [4, pp. 87])

Consider the FARE (A.314) or (A.313) defining the matrix $\bar{Q}_j$. If $\bar{Q}_j \geq 0$, $R > 0$, $[F, G]$ is stabilizable, $\left[F, \bar{Q}_j^{1/2}\right]$ is detectable, then P_j is stabilizing, i.e.

$$\bar{F}_j = F - G\left(G^T P_j G + R\right)^{-1} G^T P_j F \tag{A.315}$$

has all its eigenvalues strictly within the unit circle.

Clearly, if the conditions in ***Theorem*** A.7 are met for $j = N - 1$, then the receding horizon control law (A.304) will be stabilizing.

However, further work needs to be done to relate the design parameters, i.e. the matrices P_0, Q, R in the performance index (A.302), to the conditions in ***Theorem*** A.7. The following results from [4] can serve this purpose.

lemma **A.3** (Bitmead *et al.* [4, pp. 88])

Given two nonnegative definite symmetric matrices Q_1 and Q_2 satisfying $Q_1 \leq Q_2$ then $\left[F, Q_1^{1/2}\right]$ detectable implies $\left[F, Q_2^{1/2}\right]$ detectable.

The following corollary [5] which is an immediate result of ***lemma*** A.3 tells that if the solution of the RDE is decreasing at time j then the closed loop state transition matrix of (A.315) is stable.

Corollary A.1 (Bitmead *et al.* [5])

If the RDE with $[F, G]$ stabilizable, $\left[F, Q_1^{1/2}\right]$ detectable and if P_j in (A.312) is non-increasing at j, i.e. $P_{j+1} \leq P_j$, then $\bar{F}_j$ defined by (A.315) is stable.

Also from [5], we have the following theorem regarding the monotonicity properties of the solution of the RDE (A.312).

Theorem **A.8** (Bitmead *et al.* [5])
If the nonnegative definite solution P_j of the RDE (A.312) is monotonically non-increasing at one time, i.e. $P_{j+1} \leq P_j$ for some j, then P_j is monotonically non-increasing for all subsequent times, $P_{j+k+1} \leq P_{j+k}$, for all $k \geq 0$.

The following result is immediate by combining ***Corollary A.1*** and ***Theorem*** A.8.

Theorem **A.9** (Bitmead *et al.* [4, pp. 90])
Consider the RDE (A.312), if

- *$[F, G]$ is stabilizable*
- *$[F, Q^{1/2}]$ is detectable*
- *$P_{j+1} \leq P_j$ for some j*

then $\bar{F}_k$ given by (A.315) with P_k is stable for all $k \geq j$.

As an immediate consequence of ***Theorem*** A.9, we see that if P_0 in the design of receding horizon controller is selected in such a way that one iteration of the RDE will result in $P_1 \leq P_0$, then we have $\bar{Q}_0 = Q - (P_1 - P_0) \geq Q$ and $\bar{Q}_j \geq Q$ for any subsequent $j \geq 0$, this implies that $\bar{F}_j$ given by (A.315) is stable for any $j \geq 0$.

A clever choice of P_0 which will guarantee the monotonically non-increasing solution of RDE is to let $P_0 = \infty$ as first proposed by Kwon and Pearson [42] albeit in a very different framework. The result can be summarized as follows:

***Theorem* A.10** (Kwon *et al.*[42] and Bitmead *et al.* [4, pp. 97])
Consider system

$$x(k+1) = Fx(k) + Gu(k) \tag{A.316a}$$

$$y(k) = Hx(k) \tag{A.316b}$$

and the associated receding horizon control problem, i.e. minimize performance index

$$J(N, x(k)) = \sum_{j=0}^{N-1} \{x^T(k+j)Qx^T(k+j) + u^T(k+j)Ru(k+j)\} \tag{A.317}$$

subject to final state constraint

$$x(k+N) = 0 \tag{A.318}$$

assume $Q \geq 0$, $R > 0$, F is nonsingular and $[F, G]$ is controllable, $[F, Q]$ is observable, then the optimal solution exists and stabilizes the system (A.316) whenever $N \geq n$, where n is the dimension of system (A.316).

The nonsingularity condition of F was removed in a recent paper by Chisci and Mosca [10].

The following corollary is a natural consequence of ***Theorem*** A.10 using the argument given by Demircioglu *et al.* [16].

Corollary A.2

For system described by equation (A.316) with performance index

$$J(N, x(k)) = x^T(k+N)P_0x^T(k+N) \sum_{j=0}^{N-1} \{x^T(k+j)Qx^T(k+j) + u^T(k+j)Ru(k+j)\} \tag{A.319}$$

there exists a positive number γ such that for $P_0 > \gamma I$, the closed loop system under the control law obtained by minimizing (A.319) is also stable.

Proof: Since the pole location of the closed loop system under the optimal control law obtained by minimizing (A.319) is a continuous function of P_0, the closed loop system pole can thus be made *arbitrarily close to the limiting case of* $P_0 = \infty$ *which is stable according to* ***Theorem*** *A.10 by increasing* P_0*. Thus there always exists a positive number* γ *such that for* $P_0 > \gamma I$*, the closed loop system is stable.*

Theorem A.10 is used to investigate the stability properties of SDGPC in section 2.2.1.

References

[1] Aida, K. and T. Kitamori (1990). 'Design of a PI-type state feedback optimal servo system'. *INT. J. Control, Vol. 52, No.3.*

[2] Al-Rahmani, H. M. and G. F. Franklin (1992). 'Multirate control: A new approach'. *Automatica, Vol. 28, No. 1.*

[3] Anderson, B. D. O. and J. B. Moore (1990). *Optimal Control, Linear Quadratic Method.* Prentice Hall, Englewood Cliffs, New Jersey.

[4] Bitmead, R. R., M. Gevers and V. Wertz (1990). *Adaptive Optimal Control, The Thinking Man's GPC.* Prentice Hall.

[5] Bitmead, R. R., M. Gevers, I. R. Petersen and R. J. Kaye (1985). 'Monotonicity and stabilizablility properties of solutions of the riccati difference equation: Propositions, lemmas, theorems, fallacious conjectures and counterexamples'. *Systems and Control Letters, Vol. 5.*

[6] Bittanti, S., P. Colaneri and G. Guardabassi (1984). 'H-controllability and observability of linear periodic systems'. *SIAM Journal on Control and Optimization.*

[7] Boyd, S., L. El Ghaoui, E. Feron and V. Balakrishnan (June, 1994). *Linear Matrix Inequalities in System and Control Theory. Volume 15 of Studies in Applied Mathematics.* SIAM, Philadelphia, PA.

[8] Campo, P. J. and M. Morari (1990). 'Robust control of processes subject to saturation nonlinearities'. *Computers in chemical Engineering, Vol. 14, No. 4/5.*

[9] Chen, C. T. (1984). *Linear System Theory and Design.* New York, Holt, Rinehart and Winston.

[10] Chisci, L. and E. Mosca (September, 1993.). Stabilizing predictive control: The singular transition matrix case.. In 'Advances in Model-Based Predictive Control. Oxford, England.'.

[11] Clarke, D. W. (September,1993). Advances in model-based predictive control. In 'Workshop on Model-Based Predictive Control, Oxford University, England'.

[12] Clarke, D. W. and R. Scattolini (July 1991). 'Constrained receding-horizon predictive control'. *IEE Proceedings Vol.138 No.4.*

[13] Clarke, D. W., C. Montadi and P. S. Tuffs (1987). 'Generalized predictive control-part I. the basic algorithm.'. *Automatica, Vol.23, No.2.*

[14] Clarke, D. W., E. Mosca and R. Scattolini (December 1991). Robustness of an adaptive predictive controller. In 'Proceedings of the 30th Conference on Decision and Control,'. Brighton, England.

[15] Cutler, C. R. and B. C. Ramaker (1980). Dynamic matrix control–a computer control algorithm, paper wp5-b. In 'JACC, San Francisco'.

[16] Demircioglu, H. and D. W. Clarke (July, 1992). 'CGPC with guranteed stability properties'. *IEE Proceedings D, Vol. 139, No. 4.*

[17] Demircioglu, H. and D. W. Clarke (July, 1993). 'Generalised predictive control with end-point state weighting'. *IEE Proceedings D, Vol. 140, No. 4.*

[18] Demircioglu, H. and P. J. Gawthrop (1991). 'Continuous-time generalized predictive control (CGPC)'. *Automatica, Vol. 27, No. 1.*

[19] Demircioglu, H. and P. J. Gawthrop (1992). 'Multivariable continuous-time generalized predictive control (MCGPC)'. *Automatica.*

[20] Doyle, J. C., R. S. Smith and D. F. Enns (1987). Control of plants with input saturation nonlinearities. In '1987 ACC,'.

[21] Dumont, G. A. (1992). Fifteen years in the life of an adaptive controller. In 'IFAC Adaptive Systems in Control and Signal Processing, Grenoble, France'.

[22] Dumont, G. A. and C. C. Zervos (1986). Adaptive controllers based on orthonormal series representation. In '2nd IFAC workshop on adaptive control and signal processing'. Lund, Sweden.

[23] Dumont, G. A., Y. Fu and G. Lu (September, 1993). Nonlinear Adaptive Generalized Predictive Control and Its Applications. In 'Workshop on Model-Based Predictive Control, Oxford University, England'.

[24] Elnaggar, A., G. Dumont and A. Elshafei (December, 1990). System identification and adaptive control based on a variable regression for systems having unknown delay. In 'Proceedings of the 29th Conference on Decision and Control, Honolulu, Hawaii'.

[25] Eykhoff, P. (1974). *System Identification*. Wiley, New York.

[26] Fertik, H. A. and C. W. Ross (1967). 'Direct digital control algorithm with anti-windup feature'. *ISA Transactions, 6(4):317–328.*

[27] Finn, C. K., B. Wahlberg and B. E. Ydastie (1993). 'Constrained predictive control using orthogonal expansion'. *AIChE Journal, Vol. 39 No. 11.*

[28] Franklin, G. F., J. D. Powell and A. Emami-Naeini (1994). *Feedback Control of Dynamic Systems*. Addison-Wesley Publishing Company.

[29] Fu, Y. and G. A. Dumont (June, 1993). 'An optimal time scale for discrete Laguerre network'. *IEEE Trans. on Auto. Control, AC-38, No. 6, pp. 934–938.*

[30] Furutani, E., T. Hagiwara and M. Araki (December, 1994). Two-degree-of-freedom design method of state-predictive lqi systems. In 'Proceedings of the 33rd Conference on Decision and Control, Lake Buena Vista, FL.'.

[31] Gacía, C. E., D. M. Prett and M. Morari (1989). 'Model predictive control: Theory and practice–a survey'. *Automatica,*.

[32] Gawthrop, P. J. (1987). *Continuous-time Self-tuning Control, Volume I – Design.* Research Studies Press, England.

[33] Goodwin, G. C. and D. Q. Mayne (1987). 'A parameter estimation perspective of continuous time model reference adaptive control'. *Automatica, 23 (1), 57-70.*

[34] Hagiwara, T., T. Yamasaki and M. Araki (July, 1993*a*). Two-degree-of-freedom design method of lqi servo systems, part i: Disturbance rejection by constant feedback. In 'The 12th IFAC World Congress'.

[35] Hagiwara, T., T. Yamasaki and M. Araki (July, 1993*b*). Two-degree-of-freedom design method of lqi servo systems, part ii: Disturbance rejection by dynamic feedback. In 'The 12th IFAC World Congress'.

[36] Hanus, R., M. Kinnaert and J. L. Henrotte (1987). 'Conditioning technique, a general anti-windup and bumpless transfer method'. *Automatica, Vol. 23, 729–739.*

[37] Hautus, M. L. J. (1969). 'Controllability and observability conditions of linear autonomous systems'. *Indagationes mathematicae, Vol. 72, pp. 443–448.*

[38] Kalman, R. E., Y. C. Ho and K. S. Narendra (1963). *Contributions to Differential Equations, Vol. I.* New York: Interscience.

[39] Kothare, M., P.J. Campo and M. Morari (1993). A unified framework for the study of anti-windup designs. Technical report. CIT-CDS 93–011, California Institute of Technology.

[40] Kothare, M., V. Balakrishnan and M. Morari (1995). Robust constrained model predictive control using linear matrix inequalities. Technical report. Chemical Engineering, 210–41, California Institute of Technology.

[41] Kwakernaak, H. and R. Silvan (1960). *Linear Optimal Control Systems.* New York, Wiley-Interscience.

[42] Kwon, W. H. and A. E. Pearson (1978). 'On feedback stabilization of time-varying discrete linear system'. *IEEE Trans. AC-23, (3), pp. 479–481.*

[43] Levis, A. H., R. A. Schlueter and M. Athans (1971). 'On the behaviour of optimal linear sampled-data regulators'. *INT. J. CONTROL, Vol. 13 No. 2.*

[44] Ljung., L. (1987). *SYSTEM IDENTIFICATION: Theory for the User.* Prentice-Hall.

[45] Ljung, L. and T. Söderström (1983). *Thoery and Practice of Recursive Parameter Estimation.* MIT Press, London.

[46] Lu, G. and G. A. Dumont (Dec., 1994). Sampled-Data GPC with Integral Action: The State Space Approach. In 'Proceedings of the 33rd CDC, Lake Buena Vista, FL'.

[47] Ma, C. C. H. (December, 1991). 'Unstabilizability of linear unstable systems with input limits'. *Transactions of the ASME, Vol. 113.*

[48] Mäkilä, P. M. (1990). 'Laguerre series approximation of infinite dimensional systems'. *Automatica, Vol. 26, No. 6.*

[49] Mäkilä, P. M. (1991). 'On identification of stable systems and optimal approximation'. *Automatica, Vol. 27 No. 4.*

[50] Middleton, R. H. and G. C. Goodwin (1990). *Digital Estimation and Control: A Unified Approach.*. Englewood Cliffs, NJ: Prentice-Hall,.

[51] Morari, M. (September,1993). Model predictive control: Multivariable control technique of choice in the 1990s?. In 'Workshop on Model-Based Predictive Control, Oxford University, England'.

[52] Mosca, E. and J. Zhang (1992). 'Stable redesign of predictive control'. *Automatica, Vol. 28, No. 6, pp. 1229–1233.*

[53] Mosca, E., G. Zappa and J. M. Lemos (1989). 'Robustness of multipredictor adaptive regulators: Musmar'. *Automatica, Vol. 25 No.4.*

[54] Nicolao, G. D. and R. Scattolini (September,1993). Stability and output terminal constraints in predictive control. In 'Workshop on Model-Based Predictive Control'.

[55] Pappas, T., A. J. Laub and N. R. Sandell (1980). 'On the numerical solution of the discrete-time algebraic riccati equation'. *IEEE Trans. Auto. Control, AC-25.*

[56] Parks, T. W. (1971). 'Choice of time scale in Laguerre approximations using signal measurements'. *IEEE Trans. Auto. Control, AC–16.*

[57] Peng, H. and M. Tomizuka (1991). Preview control for vehicle lateral guidance in highway. In 'Proceedings of the 1991 American Control Conference'.

[58] Peterka, V. (1984). 'Predictor based self-tuning control'. *Automatica, Vol. 20.*

[59] Power, H. M. and B. Porter (1970, 6.). 'Necessary and sufficient conditions for controllability of multivariable systems incorporating integral feedback'. *Electron. Lett.*

[60] Rawlings, J. and K. R. Muske (1993). 'The stability of constrained receding horizon control,'. *IEEE Trans. Auto. Contr., 38.*

[61] Richalet, J., A. Rault, J. L. Testud and J. Papon (1978*a*). 'Model predictive heuristic control: Applications to Industrial Processes'. *Automatica. Vol. 14.*

[62] Richalet, J., A. Rault, J. L. Testud and J. Papon (1978*b*). 'Model predictive heuristic control: Applications to Industrial Processes'. *Automatica. Vol. 14.*

[63] Åström, K. J. and B. Wittenmark (1984). *Computer Controlled Systems-Theory and Design.* Englewood Cliffs, NJ: Prentice Hall.

[64] Åström, K. J. and B. Wittenmark (1989). *Adaptive Control.* Addison-Wesley Publishing Company.

[65] Åström, K. J. and P. Eykhoff (1971). 'System identification - a survey'. *Automatica, Vol. 7.*

[66] Robinson, W. R. and A. C. Soudak (1970). 'A method for the identification of time-delays in linear systems'. *IEEE Tran. Aut. Control.*

[67] Sagara, S. and Zhen-Yu Zhao (1990). 'Numerical integration approach to on-line identification of continuous-time systems'. *Automatica.*

[68] Salgado, M. E., G. C. Goodwin and R. H. Middleton (1988). 'Modified least squares algorithm incorporating exponential resetting and forgetting'. *Int. J. Control, Vol. 47, No. 2.*

[69] Scokaert, P. O. M. and D. W. Clarke (1994). Stability and feasibility in constrained predictive control. In 'Advances in Model Based Predictive Control'. Oxford Science Publications.

[70] Sinha, N. (1972). 'Estimation of the transfer function of a continuous-time systems from sampled data'. *IEE Proceedings Part D, Vol. 119.*

[71] Sinha, N. and S. Puthenpura (November, 1985). 'Choice of the sampling interval for the identification of continuous-time systems from samples of input/output data'. *IEE Proceedings Part D, Vol. 132(6).*

[72] Söderström, T. and P. G. Stoica (1989). *System Identification.* Prentice-Hall, Hemel Hempstead, U.K.

[73] Soeterboek, R. (1992). *Predictive Control - A Unified Approach.* Prentice-Hall.

[74] Souza, C. D., M. R. Gevers and G. C. Goodwin (Sep. 1986). 'Riccati equations in optimal filtering of nonstabilizable systems having singular state transition matrices'. *IEEE Trans. Auto. Contr. , Vol. AC-31, No. 9.*

[75] Sznaier, M. and F. Blanchini (Vol. 5, 1995). 'Robust control of constrained systems via convex optimization'. *International Journal of Robust and Nonlinear Control.*

[76] Tomizuka, M. and D. E. Whitney (Dec. 1975). 'The discrete optimal finite preview control problem (why and how is future information important?)'. *ASME Journal of Dynamic Systems, Measurement and Control, Vol. 97, No.4.*

[77] Tomizuka, M., D. Dornfeld, X. Q. Bian and H. C. Cai (Mar. 1984). 'Experimental evaluation of the preview servo scheme for a two-axis positioning system'. *ASME Journal of Dynamic Systems, Measurement and Control, Vol. 106, No.1.*

[78] Unbehauen, H. and G. P. Raö (1987). *Idnentification of Continuous Systems.* North-Holland, Amsterdam.

[79] Unbehauen, H. and G. P. Rao (1990). 'Continuous-time approaches to system identification– a survey'. *Automatica.*

[80] Wahlberg, B. (May 1991). 'System identification using Laguerre models'. *IEEE Transactions on Automatic Control, Vol. 36, No. 5.*

[81] Walgama, K. S. and J. Sternby (1990). 'Inherent observer property in a class of anti-windup compensators'. *Int. J. Control, Vol. 52, No. 3, 705–724.*

[82] Wiener, N. (1956). *The theory of Prediction. Modern Mathematics for Engineers,.* New York, McGraw-Hill.

[83] Xie, X. and R. J. Evans (1984). 'Discrete-time adaptive control for deterministic time-varying systems'. *Automatica, Vol. 20, No. 3.*

[84] Ydstie, B. (1984). Extended horizon adaptive control. In 'Proceedings of the 9th IFAC World Congress, Budapest, Hungary'.

[85] Young, P. C. (1981). 'Parameter estimation for continuous-time models– a survey.'. *Automatica, Vol. 17.*

[86] Young, P. C. and A. Jakeman (1980). 'Refined instrumental variable methods of recursive time-series analysis. part iii. extensions.'. *Int. J. Control, Vol. 31.*

[87] Zervos, C. C. and G. A. Dumont (1988). 'Deterministic adaptive control based on Laguerre series representation'. *Int. J. Control, Vol. 48, No. 6.*