DELTA-DOMAIN PREDICTIVE CONTROL AND IDENTIFICATION FOR CONTROL

Morten Bach Lauritsen

31th January 1997

Preface

This thesis has been prepared at the Department of Mathematical Modelling (IMM), the Technical University of Denmark, in partial fulfillment of the requirements for the degree of Ph.D. in engineering.

The thesis discusses aspects on modelling, control and identification of linear systems. Special attention is paid to discrete-time modelling using the delta-operator parametrization, which gives a close connection to continuous-time. Predictive control algorithms based on this parametrization are developed. Finally, aspects on the interconnection between system identification and control design are discussed.

The main contribution is the development of a unified framework for discrete-time and continuous-time predictive control algorithms.

Helsingør, January 1997

Morten Bach Lauritsen

Acknowledgements

It is with pleasure I here take the opportunity to thank a number of people who have contributed in one way or another to the present thesis.

I wish to thank my supervisor, Assoc. Prof. Niels Kjølstad Poulsen, for giving me the opportunity to work on this project. I thank the Technical University of Denmark for sponsoring the research.

During the first year of my study I had the privilege to share office with M.Sc.E, Ph.D. Morten Rostgaard. We have a common interest in many topics of the thesis and Morten has been a great source of inspiration. He has also proven to be a worthy opponent at our weekly badminton matches. I also wish to thank my office associate during the last one and a half year M.Sc.E, Ph.D. stud. Uffe Høgsbro Thygesen for many interesting discussions and his very pleasant company. I gratefully appreciate the effort Morten and Uffe have put into reading through parts of this manuscript. Their comments have certainly improved the outcome.

Finally, I wish to thank the employees at the Department of Mathematical Modelling for creating a good atmosphere and making it a pleasant place to work.

Parts of this thesis have previously been published in:

- M. B. Lauritsen (1994), "Robust Adaptive Control", Master's thesis, IMSOR, Technical University of Denmark.
- M. B. Lauritsen, M. Rostgaard and N. K. Poulsen (1994), "Emulator-based GPC in the delta-domain", Technical Report IMM-REP-1994-24, Department of Mathematical Modelling, Technical University of Denmark.
- M. B. Lauritsen, M. Rostgaard and N. K. Poulsen (1994), "Emulator-based prediction in the delta-domain", Technical Report IMM-REP-1994-25, Department of Mathematical Modelling, Technical University of Denmark.
- M. B. Lauritsen, M. Rostgaard and N. K. Poulsen (1995), "Generalized predictive control in the delta-domain", In *Proceedings of the 1995 American Control Conference*, volume 5, pages 3709–3713, Seattle, Washington, USA.
- M. B. Lauritsen, M. Rostgaard and N. K. Poulsen (1995), "Optimal prediction in the delta-domain", In *Proceedings of the 3rd European Control Conference*, volume 4, pages 2851–2856, Rome, Italy.
- M. Rostgaard, M. B. Lauritsen and N. K. Poulsen (1996), "A state space approach to the emulator-based GPC design", *Systems & Control Letters*, volume 28, pages 291–301.
- M. B. Lauritsen, M. Rostgaard and N. K. Poulsen (1996), "GPC using a delta-domain emulator-based approach", *International Journal of Control*, accepted for publication.

Summary

The present thesis is concerned with different aspects of modelling, control and identification of linear systems.

Traditionally, discrete-time sampled-data systems are represented using shift-operator parametrizations. Such parametrizations are not suitable at fast sampling rates. An alternative parametrization using the so-called delta-operator is examined. It is shown how to maintain a close correspondence to continuous-time when sampling a system described in continuous-time by stochastic differential equations. Using delta-operator parametrizations makes it possible to unify discrete-time and continuous-time theory. In addition these parametrizations possess certain numerical advantages compared to shift-operator representations.

A new prediction method is developed. It is based on ideas from continuous-time but derived from discrete-time delta-operator models. It is shown to include the optimal minimum-variance predictor as a special case and to have a well-defined continuous-time limit.

By means of this new prediction method a unified framework for discrete-time and continuous-time predictive control algorithms is developed. This contains a continuous-time like discrete-time predictive controller which is insensitive to the choice of sampling period and has a well-defined limit

in the continuous-time case. Also more conventional discrete-time predictive control methods may be described within the unified approach. The predictive control algorithms are extended to frequency weighted criterion functions. Also a state-space approach is described which extends straightforwardly to the multi-variable case.

Finally, aspects on the connection between system identification and control design are discussed. Several approaches to improve this interconnection have been proposed. The frequency-distribution of the estimation error with low-complexity models is treated and proves to be important for the development of control-relevant prefilters in estimation. Iterative approaches are presented, both using standard estimation methods with prefiltering and non-standard control-relevant estimation methods. New combined adaptive/iterative techniques are proposed.

Resumé (in Danish)

Denne afhandling omhandler forskellige aspekter vedrørende modellering, regulering og identifikation af lineære systemer.

Traditionelt set er samplede systemer blevet beskrevet ved hjælp af parametriseringer i skifte-operatoren. Sådanne parametriseringer er ikke hensigtsmæssige ved hurtig sampling. En alternativ parametrisering, der bruger den såkaldte delta-operator bliver undersøgt, og det vises hvordan en tæt sammenhæng til kontinuert tid kan opnås, ved sampling af systemer beskrevet i kontinuert tid ved stokastiske differentialligninger. Brugen af delta-operator parametriseringer gør det muligt at forene resultater for diskret og kontinuert tid. Herudover har disse parametriseringer visse numeriske fordele sammenlignet med skifte-operator repræsentationer.

En ny prædiktionsmetode bliver udviklet. Den er baseret på ideer fra kontinuert tid men udledt udfra diskrete delta-operator modeller. Det vises, at metoden indeholder optimal minimum-varians prædiktion som et special tilfælde og endvidere har en veldefineret grænseovergang til kontinuert tid.

Ved hjælp af denne nye prædiktionsmetode bliver en fælles fremstilling af diskrete og kontinuerte prædiktive reguleringsalgoritmer udviklet. Denne indeholder en kontinuert lignende diskret tids prædiktiv regulator, som er ufølsom overfor valg af samplingstid og som har en veldefineret grænseover-

gang til kontinuert tid. Også mere konventionelle diskrete prædiktive reguleringsmetoder kan beskrives indenfor den fælles fremstilling. Reguleringsalgoritmerne udvides til frekvensvægtede kriterie funktioner. Der beskrives også en fremgangsmåde baseret på modeller på tilstandsform, som let kan udvides til multivariable systemer.

Slutteligt diskuteres aspekter vedrørende sammenhængen mellem systemidentifikation og regulatordesign. Der er blevet foreslået adskillige fremgangsmåder til at forbedre denne sammenkobling. Frekvensfordelingen af estimationsfejlen med lavkompleksitets modeller bliver behandlet og viser sig at være vigtig for udviklingen af reguleringsrelevante forfiltre ved estimation. Forskellige iterative algoritmer bliver præsenteret, både med standard estimationsmetoder med forfiltrering og med specialiserede reguleringsrelevante estimationsmetoder. Nye kombinerede adaptive/iterative teknikker bliver foreslået.

Contents

Chapter 1

Introduction

Mathematical models provide an important basis for development of control systems. Often such models are derived from physical laws resulting in continuous-time descriptions. However, with the increasing use of digital computers in control and estimation it is more natural to work directly with discrete-time models. These sampled models are traditionally represented using shift-operator parametrizations. Intuitively to avoid information loss one should sample as fast as possible. Unfortunately, the traditional shift-operator models fail to provide meaningful descriptions of the sampled system at fast sampling rates. In addition it is difficult to utilize physical insight in such models. An alternative parametrization based on signal differencing rather than signal shifting has been advocated to overcome these difficulties. This so-called δ-operator parametrization has a close connection to continuous-time and in fact converges to the underlying continuous-time description as the sampling period tends to zero. Moreover, it has certain numerical advantages compared to shift-operator parametrizations.

A popular type of control is so-called long range predictive control. Such methods have proven successful in practice for several reasons. They are capable of dealing with systems with poorly known or varying time-delay. They can control non-minimumphase systems and unstable or badly damped systems. Furthermore, they provide a systematic way of dealing with hard constraints on output and input signals. To utilize the advantageous properties of δ-operator models it is interesting to formulate predictive control methods in the δ-domain.

Prediction using discrete-time models is usually focused on the sampling instants. Hence, to achieve a constant absolute prediction horizon, say $\tau = kT$, the number of predicted samples k must be increased if the sampling period T is reduced. This is a problem at fast sampling rates as the computational effort involved in finding long range predictions then increase. In particular prediction using this approach is not feasible in the continuous-time limit. To preserve a close connection to continuous-time of the predictive control algorithms alternative prediction methods must be developed. One possibility is to take starting point in methods for prediction based on continuous-time models. By formulating a discrete-time δ-domain prediction method based on continuous-time ideas it turns out to be possible to find a unified framework for discrete-time and continuous-time predictive control algorithms.

The aim of the first part of the thesis introduced above is to begin with to motivate the use of δ-operator parametrizations in discrete-time modelling. Secondly, to utilize the properties of the δ-operator to develop predictive control methods formulated in the δ-domain which unify discrete-time and continuous-time results.

The second part discusses the interconnection between system identification and control design. In modelling of physical systems it is impossible to avoid systematic modelling errors not attributable to noise, so-called bias. Traditionally, system identification and control design have existed as separate disciplines. The model estimate is most often not accompanied by error bounds and the control design is often solely based on a nominal model. This is unfortunate, as the closed-loop performance may be severely deteriorated even by small modelling errors if these are not taken into account. Hence, it is necessary to take certain measures to reduce the effect of modelling errors. There are two principal ways to attack this problem. The first is to develop identification methods, which beside a nominal model, produce bounds on the modelling errors to be used in robust control design. The other is to modify the identification algorithm to take into account the modelling error in such a way, that a nominal model suitable for high performance control is estimated. This is the approach pursued here.

The estimation of a model and a subsequent model-based control design can be viewed as a joint optimization problem. However, such problems are rarely tractable in practice. To overcome this several iterative procedures for combined system identification and control design have been proposed. With a given controller it is possible, by comparing the achieved closed-loop with the designed closed-loop, to set up a truly control-relevant identification problem derived directly from the control criterion. System identification can then be interpreted as a way to reduce the performance degradation caused by the use of a model rather than the true system in control design. The idea of the iterative procedures is then to use successive steps of local estimation and control problems, i.e. model estimation with fixed controller and control design with a fixed model.

The aim of the second part of the thesis is to give an overview of current research in control-relevant identification and iterative methods along with the theoretical background. New interpretations are given and new combined adaptive/iterative techniques are proposed. The methods under consideration are presented with a common notation for the sake of readability.

1.1 Outline of the thesis

Chapter 2 discusses modelling of linear systems. Discrete-time models are derived by sampling systems described in continuous-time as stochastic differential equations. Two different discrete-time operators are considered—the shift-operator q and the delta-operator δ. The limiting properties of the discrete-time models as the sampling period tends to zero are investigated. Some aspects of estimation using δ-operator parametrizations are considered. Finally, some results on the superior numerical properties of δ-domain models as compared to q-domain models are reviewed.

In Chapter 3 methods for long range prediction are considered. Optimal minimum variance predictions for both q-domain and δ-domain models are found and recursive algorithms for their computation are given. A new so-called δ-domain emulator-based predictor is developed based on discrete-time models. Nevertheless, it is able to predict the output continuously and has a well-defined limit as the sampling period tends to zero.

The use of these prediction methods in long range predictive control design is investigated in Chapter 4. First an introduction to predictive control is given via a brief description of the popular Generalized Predictive Controller. An overview of research on extensions to the basic GPC is given. After that, new predictive control methods formulated in the δ-domain are introduced. A framework is developed which provides a unification of discrete-time and continuous-time predictive control. A number of extensions to the basic algorithms are given. This includes general frequency weighted criterion functions and a new future control constraint. Last a state-space formulation of the basic δ-domain controllers is developed. This formulation extends easily to the multi-variable case. Two such extensions are presented.

The second part of the thesis is comprised by Chapter 5. This chapter is concerned with methods to improve the connection between identification and control and can be read independently from the first part. Some basic results on the frequency-distribution of bias in transfer function estimation are reviewed, both for open-loop and closed-loop systems. These results are important in the development of control-relevant prefilters in estimation. An optimal choice of prefilter is found and other proposed prefiltering designs are described. Subsequently, attention is turned to iterative identification and control algorithms. A motivation is given for control-relevant identification problems derived directly from the control criterion. Finally, new combined adaptive/iterative approaches using on-line estimation are proposed.

Chapter 2

Modelling of Linear Systems

2.1 Introduction

In this chapter some aspects of linear discrete-time modelling will be discussed. We investigate how discrete-time models are derived by sampling of continuous-time processes and how input-output descriptions relate to state-space models. Two different discrete-time parametrizations are considered. The traditional shift-operator parametrization is shown to have some difficulties at fast sampling rates—contrary to common expectations such parametrizations do not converge to their continuous-time counterparts as the sampling period tends to zero.

For that reason, a great deal of attention is given to the alternative so-called δ-operator parametrization which uses signal differencing rather than

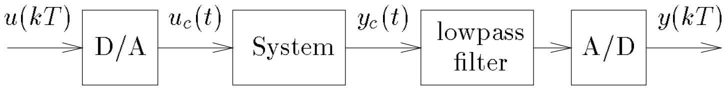

Figure 2.1: Structure of sampled-data systems.

shifting. These δ-operator parametrizations show a close correspondence to continuous-time descriptions and in fact they converge to the expected continuous-time equivalent as the sampling period tends to zero.

2.2 Sampled-data systems

The use of digital computers in implementation of process control systems has become very widespread. The discrete nature of digital computers makes it convenient to use discrete-time models of the controlled processes. Consider a sampled data system with the principal structure of Figure 2.1. The discrete-time input $u(kT)$, where T is the sampling period and k is an integer, is processed by a digital to analog (D/A) converter to give the continuous-time input $u_c(t)$. Usually, the D/A-converter is designed in such a way, that the value of $u_c(t)$ is held constant between samples, known as zero-order-hold (ZOH). The continuous-time output $y_c(t)$ is then sampled with a period T using an analog to digital (A/D) converter to give the sampled output $y(kT)$. In practice one must prefilter the continuous-time output before sampling to avoid aliasing problems. The prefilter should be of low-pass nature with a break frequency around the Nyquist-frequency π/T. This is discussed in more detail in Section 2.4.6. In the following we will neglect the effect of this prefilter.

It should be noted, that from now on the variable t is used as the time-variable in both continuous-time and discrete-time. In discrete-time t can only assume values, that are multiples of the sampling period T, i.e. $t = kT$ where k is an integer. The distinction should be clear from the context.

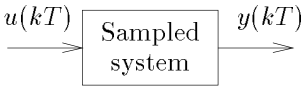

Figure 2.2: Sampled system description.

The class of systems under consideration in this thesis is limited to linear finite-dimensional time-invariant systems. If not otherwise mentioned we also assume the system to be single-input single-output (SISO).

When working with digital control systems it is convenient to find an equivalent discrete-time description of the composite system consisting of D/A-converter, continuous-time system and A/D-converter, i.e. to find a direct relation between the sampled signals $u(kT)$ and $y(kT)$ as illustrated in Figure 2.2. This is the subject of the next two sections. We consider the cases of deterministic and stochastic system descriptions separately.

2.3 Sampling of deterministic systems

Consider the deterministic time-invariant continuous-time linear state-space model

$$\dot{x}(t) = \boldsymbol{A}_c x(t) + \boldsymbol{B}_c u(t) \tag{2.1}$$

$$y(t) = \boldsymbol{C}_c x(t) \tag{2.2}$$

The order of the system is n, that is $\dim(\boldsymbol{A}_c) = n \times n$.

We consider sampling with a zero-order hold on the input, i.e. the input-signal is held constant between sampling instants. This is common in digital control and is easily implemented by simply using a D/A-converter, which latches the signal between sampling instants. Given a state value $x(kT)$ at some sampling point kT, where T is the sampling period and k is an integer, the state evolution is

$$x(t) = e^{\boldsymbol{A}_c(t-kT)}x(kT) + \int_{kT}^{t} e^{\boldsymbol{A}_c(t-\tau)}\boldsymbol{B}_c u(\tau)\,d\tau \tag{2.3}$$

where we have assumed right continuity of the control signal. Hence, at the next sampling instant the state is given by

$$\begin{aligned} x(kT+T) &= e^{\boldsymbol{A}_c T}x(kT) + \int_{kT}^{kT+T} e^{\boldsymbol{A}_c(kT+T-\tau)}\boldsymbol{B}_c u(kT)\,d\tau \\ &= e^{\boldsymbol{A}_c T}x(kT) + \int_{kT}^{kT+T} e^{\boldsymbol{A}_c(kT+T-\tau)}\,d\tau \boldsymbol{B}_c u(kT) \\ &= e^{\boldsymbol{A}_c T}x(kT) + \int_{0}^{T} e^{\boldsymbol{A}_c \tau}\,d\tau \boldsymbol{B}_c u(kT) \end{aligned} \tag{2.4}$$

From this we see, that a discrete-time state-space description is

$$\begin{aligned} qx(t) &= \boldsymbol{A}_q x(t) + \boldsymbol{B}_q u(t) \\ y(t) &= \boldsymbol{C}_q x(t) \end{aligned} \tag{2.5}$$

where t must be a multiple of the sampling period T and

$$\boldsymbol{A}_q = e^{\boldsymbol{A}_c T} \tag{2.6}$$

$$\boldsymbol{B}_q = \int_0^T e^{\boldsymbol{A}_c \tau}\,d\tau \boldsymbol{B}_c \tag{2.7}$$

$$\boldsymbol{C}_q = \boldsymbol{C}_c \tag{2.8}$$

The operator q is the forward-shift operator, that is $qx(t) = x(t+T)$. It is important to observe, that if the real continuous-time control signal $u(t)$ is in fact constant between the sampling instants then the sampling involves no approximation. In this case, the difference equation (2.5) gives the exact value of the state and the output at the sampling instants. This is of course only valid for systems without noise.

An equivalent shift-operator input-output model on the form

$$A_q(q)y(t) = B_q(q)u(t) \tag{2.9}$$

where $A_q(q)$ and $B_q(q)$ are polynomials in the forward shift-operator can be found from the following rewritings

$$\begin{aligned} y(t) &= \boldsymbol{C}_q x(t) \\ &= \boldsymbol{C}_q \left(q\boldsymbol{I} - \boldsymbol{A}_q\right)^{-1} \boldsymbol{B}_q u(t) \\ &= \boldsymbol{C}_q \frac{\mathrm{adj}(q\boldsymbol{I} - \boldsymbol{A}_q)}{\det(q\boldsymbol{I} - \boldsymbol{A}_q)} \boldsymbol{B}_q u(t) \end{aligned} \tag{2.10}$$

where $\boldsymbol{I}$ is a unity matrix of appropriate dimension and $\mathrm{adj}(\cdot)$ is the adjoint matrix. Hence,

$$\det(q\boldsymbol{I} - \boldsymbol{A}_q) y(t) = \boldsymbol{C}_q \,\mathrm{adj}(q\boldsymbol{I} - \boldsymbol{A}_q) \boldsymbol{B}_q u(t) \tag{2.11}$$

from where we identify

$$A_q(q) = \det(q\boldsymbol{I} - \boldsymbol{A}_q) \tag{2.12}$$

$$B_q(q) = \boldsymbol{C}_q \,\mathrm{adj}(q\boldsymbol{I} - \boldsymbol{A}_q) \boldsymbol{B}_q \tag{2.13}$$

where in general

$$\deg(A_q) = n \tag{2.14}$$

$$\deg(B_q) = m \leq n - 1 \tag{2.15}$$

The delay d in samples of the discrete-time description is given implicitly as the relative degree of $B_q(q)/A_q(q)$, that is $d = n - m \geq 1$.

2.4 Sampling of stochastic systems

Deterministic systems are of course rare in the real world. Typically the system will be subject to noise from various sources. Often this is modelled by adding a state-excitation noise vector to the state equation and a measurement noise vector to the observation equation. We will pay some attention to the mathematical modelling of these noise contributions in continuous-time models and how this is carried over to the discrete-time descriptions.

Also the derivation of equivalent input-output descriptions for stochastic models requires results from linear filtering theory. These results will be briefly recapitulated. They also help us understand the connection to continuous-time of the discrete-time models considered in this chapter. In particular, they emphasize the short-comings of discrete-time q-domain descriptions at fast sampling rates. The results are standard and can be found in e.g. (Kwakernaak and Sivan 1972, Jazwinski 1970, Åström 1970, Middleton and Goodwin 1990, Goodwin and Sin 1984) or in the original work by Kalman. The filtering results hold for multi-input multi-output (MIMO) systems. However, we only consider transfer function representations for SISO systems.

2.4.1 Continuous-time system description

Consider the stochastic continuous-time state-space description

$$dx_c(t) = \boldsymbol{A}_c x_c(t)\,dt + \boldsymbol{B}_c u(t)\,dt + dv_c(t) \tag{2.16}$$

$$dz_c(t) = \boldsymbol{C}_c x_c(t)\,dt + de_c(t) \tag{2.17}$$

where the state noise process $\{dv_c(t)\}$ and the measurement noise process $\{de_c(t)\}$ are mutually independent Wiener processes (also known as Brownian motion processes) with incremental covariances

$$\mathcal{E}\{dv_c(t)dv_c(t)^\top\} = \boldsymbol{\Omega}_c dt \tag{2.18}$$

$$\mathcal{E}\{de_c(t)de_c(t)^\top\} = \boldsymbol{\Gamma}_c dt \tag{2.19}$$

The so-called intensities or spectral densities $\boldsymbol{\Omega}_c$ and $\boldsymbol{\Gamma}_c$ are positive definite. Both the state-vector $x_c(t)$ and the output process $z_c(t)$ are described by stochastic differential equations. Formally, the state-space equations may be written as

$$\begin{aligned} \dot{x}_c(t) &= \boldsymbol{A}_c x_c(t) + \boldsymbol{B}_c u(t) + \dot{v}_c(t) \\ y_c(t) &= \boldsymbol{C}_c x_c(t) + \dot{e}_c(t) \end{aligned} \tag{2.20}$$

where $y_c(t) = \dot{z}_c(t)$ and $\dot{v}_c(t)$ and $\dot{e}_c(t)$ are independent continuous-time white noise processes with covariance functions

$$\mathcal{E}\{\dot{v}_c(t)\dot{v}_c(\tau)^\top\} = \boldsymbol{\Omega}_c i(t-\tau) \tag{2.21}$$

$$\mathcal{E}\{\dot{e}_c(t)\dot{e}_c(\tau)^\top\} = \boldsymbol{\Gamma}_c i(t-\tau) \tag{2.22}$$

where $i(t-\tau)$ is the continuous-time Dirac delta-function. The variances given in (2.21) and (2.22) are infinite and thus the processes $\dot{v}_c(t)$ and $\dot{e}_c(t)$ have no physical interpretation. In practice, the noise signals acting on the state and the measured output will be of limited bandwidth. The non-physical system description shown above can easily be extended to the case of colored noise simply by augmenting the state variable, (Kwakernaak and Sivan 1972). The use of continuous-time white noise is convenient from a theoretical viewpoint, for instance in the derivation of the optimal continuous-time filter which is considered next.

2.4.2 Continuous-time optimal filtering

In order to evaluate the properties of the discrete-time models in the continuous-time limit as the sampling period tends to zero we briefly recapitulate some results from linear filtering theory. The following theorem gives the solution to the non-singular continuous-time optimal observer problem.

Theorem 1 *Consider the continuous-time system*

$$\begin{aligned}\dot{x}_c(t) &= \boldsymbol{A}_c x_c(t) + \boldsymbol{B}_c u(t) + \dot{v}_c(t)\\ y_c(t) &= \boldsymbol{C}_c x_c(t) + \dot{e}_c(t) \qquad t \geq t_0\end{aligned} \tag{2.23}$$

with initial state $x_c(t_0) \in N(m_0, \boldsymbol{P}_0)$ and independent of $\dot{v}_c(t)$ and $\dot{e}_c(t)$. The noise sequences $\dot{v}_c(t)$ and $\dot{e}_c(t)$ are mutually independent zero-mean white noise with constant intensities $\boldsymbol{\Omega}_c$ and $\boldsymbol{\Gamma}_c > 0$, respectively.

The optimal observer which minimizes for all $t \geq t_0$

$$\mathcal{E}\left\{[x_c(t) - \hat{x}_c(t)]^\top [x_c(t) - \hat{x}_c(t)]\right\} \tag{2.24}$$

is given by

$$\begin{aligned} \dot{\hat{x}}_c(t) &= \boldsymbol{A}_c\hat{x}_c(t) + \boldsymbol{B}_c u(t) + \boldsymbol{K}_c(t)\tilde{y}_c(t) \\ \tilde{y}_c(t) &= y_c(t) - \boldsymbol{C}_c\hat{x}_c(t) \end{aligned} \tag{2.25}$$

where the gain matrix $\boldsymbol{K}_c(t)$ *is chosen as*

$$\boldsymbol{K}_c(t) = \boldsymbol{P}_c(t)\boldsymbol{C}_c^\top \boldsymbol{\Gamma}_c^{-1} \tag{2.26}$$

with $\boldsymbol{P}_c(t)$ *being the solution to the matrix Riccati equation*

$$\dot{\boldsymbol{P}}_c(t) = \boldsymbol{A}_c\boldsymbol{P}_c(t) + \boldsymbol{P}_c(t)\boldsymbol{A}_c^\top + \boldsymbol{\Omega}_c - \boldsymbol{K}_c(t)\boldsymbol{\Gamma}_c\boldsymbol{K}_c(t)^\top \tag{2.27}$$

with initial conditions

$$\boldsymbol{P}_c(t_0) = \boldsymbol{P}_0 \tag{2.28}$$

$$\hat{x}_c(t_0) = m_0 \tag{2.29}$$

The mean square reconstruction error of the state estimate is

$$\mathcal{E}\left\{[x_c(t) - \hat{x}_c(t)][x_c(t) - \hat{x}_c(t)]^\top\right\} = \boldsymbol{P}_c(t) \tag{2.30}$$

□

This is the well-known Kalman-Bucy or Kalman filter. The gain matrix $\boldsymbol{K}_c(t)$ is known as the Kalman gain. It can be shown, that with Gaussian noise no other function, linear or nonlinear, of the input and output signals can provide an estimate with a smaller mean square reconstruction error, (Kwakernaak and Sivan 1972).

Steady-state filter

Under mild restrictions the optimal filter converges as $t_0 \to -\infty$ to a steady-state solution independent of the initial conditions. The matrix Riccati equation (2.27) then converges to an algebraic relation. This is summarized in the following theorem.

Theorem 2 *Consider the optimal filter of Theorem 1. Factor the state-noise intensity as*

$$\boldsymbol{\Omega}_c = \boldsymbol{\Omega}_c^{1/2}(\boldsymbol{\Omega}_c^{1/2})^\top \tag{2.31}$$

Then if $(\boldsymbol{C}_c, \boldsymbol{A}_c)$ *is detectable and* $(\boldsymbol{A}_c, \boldsymbol{\Omega}_c^{1/2})$ *is reachable the Kalman-Bucy filter converges for* $t_0 \to -\infty$ *to a stable steady-state filter given by*

$$\begin{aligned} \dot{\hat{x}}_c(t) &= \boldsymbol{A}_c\hat{x}_c(t) + \boldsymbol{B}_c u(t) + \boldsymbol{K}_c\tilde{y}_c(t) && (2.32) \\ \tilde{y}_c(t) &= y_c(t) - \boldsymbol{C}_c\hat{x}_c(t) && (2.33) \end{aligned}$$

where $\boldsymbol{P}_c$ *is the unique positive definite solution to the algebraic observer Riccati equation*

$$0 = \boldsymbol{A}_c\boldsymbol{P}_c + \boldsymbol{P}_c\boldsymbol{A}_c^\top + \boldsymbol{\Omega}_c - \boldsymbol{K}_c\boldsymbol{\Gamma}_c\boldsymbol{K}_c^\top \tag{2.34}$$

and the steady-state Kalman gain is

$$\boldsymbol{K}_c = \boldsymbol{P}_c\boldsymbol{C}_c^\top\boldsymbol{\Gamma}_c^{-1} \tag{2.35}$$

The stationary innovation process $\tilde{y}(t)$ *is a continuous-time white noise process with intensity* $\boldsymbol{\Gamma}_c$*, i.e.*

$$\mathcal{E}\{\tilde{y}(t)\tilde{y}(t)^\top\} = \boldsymbol{\Gamma}_c i(0) \tag{2.36}$$

□

In practice, the convergence rate of the time-varying filter to its steady-state equivalent is often quite fast, depending on poles of the filter. Hence, the steady-state filter, which is time-invariant, is attractive to use as it is much easier to implement.

2.4.3 Sampled description

We want to find a description of the discrete-time model obtained when the continuous-time stochastic system (2.20) is sampled with period T.

For the state-equation part of the model one gets the discrete-time shift-operator state-space equation by Itô-integration of the continuous-time equation. The calculations are analogue to the deterministic case except for the presence of the state-excitation noise. We obtain the following discrete-time description

$$qx(t) = \boldsymbol{A}_q x(t) + \boldsymbol{B}_q u(t) + v(t) \tag{2.37}$$

where t is a multiple of the sampling period and

$$\boldsymbol{A}_q = e^{\boldsymbol{A}_c T} \tag{2.38}$$

$$\boldsymbol{B}_q = \int_0^T e^{\boldsymbol{A}_c \tau}\, d\tau \boldsymbol{B}_c \tag{2.39}$$

$$v(kT) = \int_{kT}^{kT+T} e^{\boldsymbol{A}_c (kT+T-\tau)}\, dv_c(\tau) \tag{2.40}$$

The equivalent discrete-time white noise sequence $v(kT)$ is obtained as a Wiener integral of the Wiener process $dv_c(\tau)$, (Jazwinski 1970). As in the deterministic case it has been assumed, that the input signal $u(t)$ is constant between sampling instants.

The covariance of the state noise is given by (Åström and Wittenmark 1990, Jazwinski 1970)

$$\begin{aligned}
\mathcal{E}\{v(kT)v(kT)^\top\} &= \\
\mathcal{E}\left\{\int_{kT}^{kT+T}\int_{kT}^{kT+T} e^{\boldsymbol{A}_c(kT+T-\tau)}\, dv_c(\tau)\, dv_c(t)^\top e^{\boldsymbol{A}_c^\top (kT+T-t)}\right\} \\
&= \int_{kT}^{kT+T} e^{\boldsymbol{A}_c(kT+T-\tau)} \boldsymbol{\Omega}_c e^{\boldsymbol{A}_c^\top(kT+T-\tau)}\, d\tau \\
&= \int_0^T e^{\boldsymbol{A}_c\tau}\boldsymbol{\Omega}_c e^{\boldsymbol{A}_c^\top \tau}\, d\tau &(2.41)\\
&\triangleq \boldsymbol{Q}_q &(2.42)
\end{aligned}$$

It appears, for instance by making a Taylor-series expansion of $e^{\boldsymbol{A}_c\tau}$, that for small sampling periods T we have the following approximation

$$\boldsymbol{Q}_q \approx T\boldsymbol{\Omega}_c \tag{2.43}$$

That is, for fast sampling $\boldsymbol{Q}_q$ is of order T. Actually, as noted in (Farrell and Livstone 1996) the integral (2.41) can be solved explicitly to give

$$\boldsymbol{Q}_q = \boldsymbol{P} - e^{\boldsymbol{A}_c T} \boldsymbol{P} e^{\boldsymbol{A}_c T} \tag{2.44}$$

where $\boldsymbol{P}$ fulfills the Lyapunov equation $\boldsymbol{A}_c \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_c^\top = -\boldsymbol{\Omega}_c$ for stable systems and $\boldsymbol{A}_c \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_c^\top = \boldsymbol{\Omega}_c$ for anti-stable systems.

It is not possible to sample the continuous-time output $y_c(t)$ directly as this would yield a discrete-time description with measurement noise of infinite variance. However, in practice the measuring device and/or the A/D-converter will have a low-pass filtering effect on the measured output due to bandwidth limitations. In fact, as mentioned in Section 2.2, one should use an anti-aliasing filter of low-pass nature as part of the sampling process. This will be discussed in more detail in Section 2.4.6. With such low-pass filtering of the measured output the system can be sampled directly. (Middleton and Goodwin 1990) shows, that with a typical presampling filter, for instance a simple reset and integrate circuit, then at fast sampling rates, the discrete-time output can be expressed as

$$y(t) = \boldsymbol{C}_q x(t) + e(t) \tag{2.45}$$

where $e(t)$ is a discrete-time white noise sequence with variance $\boldsymbol{R}_q$ and for small T

$$\begin{aligned} \boldsymbol{C}_q &\approx \boldsymbol{C} && (2.46)\\ \boldsymbol{R}_q &\approx \frac{1}{T}\boldsymbol{\Gamma}_c && (2.47)\\ \mathcal{E}\{v(t)e(s)^\top\} &\approx 0 && (2.48) \end{aligned}$$

Note, that for small sampling periods the variance of the measurement noise is of order $1/T$. Hence, the variance increases towards infinity as $T \to 0$.

2.4.4 Discrete-time optimal filtering

The discrete-time version of the optimal observer considered in Theorem 1 is given in following theorem which can be found in e.g. (Kwakernaak and Sivan 1972).

Theorem 3 *Consider the sampled system description*

$$\begin{aligned} qx(t) &= \boldsymbol{A}_q x(t) + \boldsymbol{B}_q u(t) + v(t) \\ y(t) &= \boldsymbol{C}_q x(t) + e(t) \qquad t = kT,\ k \geq k_0 \end{aligned} \tag{2.49}$$

with initial state $x(k_0T) \in N(m_0, \boldsymbol{P}_0)$ independent of $v(t)$ and $e(t)$. The noise sequences $v(t)$ and $e(t)$ are mutually independent discrete-time white noise processes with variance $\boldsymbol{Q}_q$ and $\boldsymbol{R}_q > 0$, respectively.

The discrete-time optimal observer which minimizes for all $t = kT,\ k \geq k_0$

$$\mathcal{E}\left\{[x(t) - \hat{x}(t)]^\top [x(t) - \hat{x}(t)]\right\} \tag{2.50}$$

is given by

$$\begin{aligned} q\hat{x}(t) &= \boldsymbol{A}_q \hat{x}(t) + \boldsymbol{B}_q u(t) + \boldsymbol{K}_q(t)\tilde{y}(t) \\ \tilde{y}(t) &= y(t) - \boldsymbol{C}_q \hat{x}(t) \end{aligned} \tag{2.51}$$

where the gain matrix $\boldsymbol{K}_q(t)$ is chosen as

$$\boldsymbol{K}_q(t) = \boldsymbol{A}_q \boldsymbol{P}_q(t) \boldsymbol{C}_q^\top \left(\boldsymbol{C}_q \boldsymbol{P}_q(t) \boldsymbol{C}_q^\top + \boldsymbol{R}_q\right)^{-1} \tag{2.52}$$

with $\boldsymbol{P}_q(t)$ being the solution to the matrix Riccati difference equation (RDE)

$$q\boldsymbol{P}_q(t) = \boldsymbol{A}_q \boldsymbol{P}_q(t) \boldsymbol{A}_q^\top - \boldsymbol{K}_q(t)\left(\boldsymbol{C}_q \boldsymbol{P}_q(t) \boldsymbol{C}_q^\top + \boldsymbol{R}_q\right) \boldsymbol{K}_q(t)^\top + \boldsymbol{Q}_q \tag{2.53}$$

with initial conditions

$$\boldsymbol{P}_q(k_0T) = \boldsymbol{P}_0 \tag{2.54}$$

$$\hat{x}(k_0T) = m_0 \tag{2.55}$$

The variance on the error of the state estimate is

$$\mathcal{E}\left\{[x(t)-\hat{x}(t)][x(t)-\hat{x}(t)]^{\top}\right\} = \boldsymbol{P}_q(t), \qquad t = kT \tag{2.56}$$

□

The above theorem gives the discrete-time Kalman predictor. This predictor can also be interpreted as the conditional mean of the state $x(t)$ given input-output data up to time $t-T$, that is (Åström and Wittenmark 1990)

$$\hat{x}(t) = \hat{x}(t|t-T) \triangleq \mathcal{E}\{x(t)|\mathcal{I}_{i,t}\} \tag{2.57}$$

where the information set is given by

$$\mathcal{I}_{i,t} = \{y(t-T), y(t-2T), \ldots, u(t-T), u(t-2T), \ldots\} \tag{2.58}$$

Often also the latest measurement $y(t)$ is available in the reconstruction of the state giving rise to the Kalman filter rather than the Kalman predictor. This can be solved by expressing the output $y(t)$ in terms of $x(t-T)$ and using the solution to the prediction problem, see (Kwakernaak and Sivan 1972).

2.4.5 Transfer function representations

In input-output representations, for example ARMAX descriptions, we only have one driving noise signal which lumps together the effect of state excitation and measurement noise. In addition such input-output relations only describe the stationary behaviour of the system. For this reason it is necessary to rewrite the state-space model to the stationary innovations form to find transfer function representations.

Theorem 4 *Consider the optimal discrete-time filter of Theorem 3. Factor the state-noise variance as*

$$\boldsymbol{Q}_q = \boldsymbol{Q}_q^{1/2}(\boldsymbol{Q}_q^{1/2})^{\top} \tag{2.59}$$

Then if $(\boldsymbol{C}_q, \boldsymbol{A}_q)$ *is detectable and* $(\boldsymbol{A}_q, \boldsymbol{Q}_q^{1/2})$ *is reachable the stationary innovations representation exists and is given by*

$$\begin{aligned} qx(t) &= \boldsymbol{A}_q x(t) + \boldsymbol{B}_q u(t) + \boldsymbol{K}_q \tilde{y}(t) && (2.60)\\ y(t) &= \boldsymbol{C}_q x(t) + \tilde{y}(t) && (2.61) \end{aligned}$$

where $\boldsymbol{P}_q$ *is the unique positive definite solution to the discrete-time q-domain Algebraic Riccati Equation (ARE)*

$$\boldsymbol{P}_q = \boldsymbol{A}_q \boldsymbol{P}_q \boldsymbol{A}_q^\top - \boldsymbol{K}_q \left(\boldsymbol{C}_q \boldsymbol{P}_q \boldsymbol{C}_q^\top + \boldsymbol{R}_q \right) \boldsymbol{K}_q^\top + \boldsymbol{Q}_q \tag{2.62}$$

and the Kalman gain is

$$\boldsymbol{K}_q = \boldsymbol{A}_q \boldsymbol{P}_q \boldsymbol{C}_q^\top \left(\boldsymbol{C}_q \boldsymbol{P}_q \boldsymbol{C}_q^\top + \boldsymbol{R}_q \right)^{-1} \tag{2.63}$$

The innovation process $\tilde{y}(t)$ *is a white noise sequence with variance*

$$\mathcal{E}\{\tilde{y}(t)\tilde{y}(t)^\top\} = \boldsymbol{C}_q \boldsymbol{P}_q \boldsymbol{C}_q^\top + \boldsymbol{R}_q \tag{2.64}$$

□

The algebraic Riccati equation (2.62) is the stationary version of the Riccati difference equation (2.53) obtained as $t_0 \to -\infty$. The ARE is produced from the RDE by noting that in stationarity $P_q(kT + T) = P_q(kT)$.

Using the innovations representation we get in the single-input single-output case

$$y(t) = \boldsymbol{C}_q \left(q\boldsymbol{I} - \boldsymbol{A}_q \right)^{-1} \left[\boldsymbol{B}_q u(t) + \boldsymbol{K}_q \tilde{y}(t) \right] + \tilde{y}(t) \tag{2.65}$$

which leads to

$$\begin{aligned} \det(q\boldsymbol{I} - \boldsymbol{A}_q) y(t) &= \boldsymbol{C}_q \operatorname{adj}(q\boldsymbol{I} - \boldsymbol{A}_q) \boldsymbol{B}_q u(t) \\ &\quad + \left(\boldsymbol{C}_q \operatorname{adj}(q\boldsymbol{I} - \boldsymbol{A}_q) \boldsymbol{K}_q + \det(q\boldsymbol{I} - \boldsymbol{A}_q) \right) \tilde{y}(t) \end{aligned}$$

This is an input-output relation of the ARMAX form

$$A_q(q) y(t) = B_q(q) u(t) + C_q(q) \tilde{y}(t) \tag{2.66}$$

with

$$
\begin{aligned}
A_q(q) &= \det(q\boldsymbol{I} - \boldsymbol{A}_q) && (2.67)\\
B_q(q) &= \boldsymbol{C}_q \operatorname{adj}(q\boldsymbol{I} - \boldsymbol{A}_q)\boldsymbol{B}_q && (2.68)\\
C_q(q) &= \boldsymbol{C}_q \operatorname{adj}(q\boldsymbol{I} - \boldsymbol{A}_q)\boldsymbol{K}_q + \det(q\boldsymbol{I} - \boldsymbol{A}_q) && (2.69)
\end{aligned}
$$

In fact, such an ARMAX description can be considered a short-hand notation for the stationary innovation form. In general the polynomial degrees are

$$
\begin{aligned}
\deg(A_q) &= n && (2.70)\\
\deg(B_q) &\leq n-1 && (2.71)\\
\deg(C_q) &= n && (2.72)
\end{aligned}
$$

Note, that $A_q(q)$ is monic. Also when working with input-output descriptions one can without loss of generality assume, that the noise polynomial $C_q(q)$ is monic as this can be obtained by adjusting the variance of the driving noise signal. The time delay of the transfer function in samples is given implicitly as the relative degree of $B_q(q)/A_q(q)$. Usually, the discrete-time description will have a time-delay of one sample giving rise to $n-1$ discrete-time zeros independent of the number of continous-time zeros. This will be investigated further in Section 2.6.

2.4.6 Selection of sampling rate

The selection of sampling rate is an important issue in sampled data systems. The sampling process will unavoidably result in some loss of information and intuitively one would then suggest to sample as fast as possible. This is however not always a good idea. In fact the traditional q-domain parametrization of a discrete-time system fails to converge to the continuous-time equivalent as T tends to zero. A brief discussion on some aspects in the selection of sampling rate is given in the following.

General guidelines for the selection of sampling rate are discussed in (Åström and Wittenmark 1990). A common rule of thump given there, is to choose the sampling frequency as 4–10 times the desired bandwidth of the system. Too slow sampling gives a significant information loss which usually results in poorer control performance. On the other hand too fast sampling gives numerical problems, especially with discrete-time parametrization in the traditional shift-operator. (Middleton and Goodwin 1990) argues, that in modern control systems the sampling frequency should be 10–50 times the closed-loop bandwidth.

Anti-aliasing filters

Consider a continuous-time signal $s(t)$ with two-sided spectrum

$$\Phi_c(\omega) \tag{2.73}$$

If sampled with a sampling frequency of $\omega_s = 2\pi/T$ where T is the sampling period, the spectrum of the sampled signal becomes (Ljung 1987)

$$\Phi_T(\omega) = \sum_{i=-\infty}^{\infty} \Phi_c(\omega + i\,\omega_s) \tag{2.74}$$

i.e. the spectrum of the sampled signal is obtained as a superposition of the original spectrum repeated at multiples of the sampling frequency. This introduces a possibility for the so-called *aliasing* phenomenon: If the continuous-time signal has a higher bandwidth than the Nyquist-frequency $\omega_N = \omega_s/2$ then the spectrum of the discrete-time signal will be distorted.

To avoid aliasing it is necessary to use *anti-aliasing filters*, that is a low-pass filtering of the continuous-time signal before the actual sampling takes place; preferably by an ideal low-pass filter with break-frequency $\omega_s/2$. If the noise-contribution in the continuous-time system is wide-band then the anti-aliasing filter will have a noise-reduction effect. Suppose the stationary continuous-time signal $v(t)$ has spectrum $\Phi_c^v(\omega)$. The variance of the sampled signal $v(kT)$ is then given by

$$E\{v(kT)^2\} = \int_{-\omega_s/2}^{\omega_s/2} \sum_{i=-\infty}^{\infty} \Phi_c^v(\omega + i\,\omega_s)\,d\omega \qquad (2.75)$$

This shows, that if the continuous-time signal has energy above the Nyquist-frequency then this energy is folded down in the low-frequency area, which increases the variance of the discrete-time signal. However, if $v(t)$ is ideally low-pass filtered before the sampling takes place giving the signal $v^f(t)$ then the variance of the corresponding discrete-time signal $v_f(kT)$ is given by

$$E\{v_f(kT)^2\} = \int_{-\omega_s/2}^{\omega_s/2} \Phi_c^v(\omega)\,d\omega \qquad (2.76)$$

which may be a significant noise reduction if $v(t)$ has large high-frequency contributions. Apparently, using slow sampling with a suitable anti-aliasing filter has a noise reducing effect. However, using slow sampling will also give little information about the system dynamics as all contributions in the output of frequency higher than the Nyquist-frequency are filtered away.

In digital control applications the input signal is usually applied to the process through a zero-order hold, i.e. the value of the input signal is held constant between sampling instants. This gives rise to stepwise changes of the input-signal at the sampling instants. These steps contain high frequencies which will excite the high frequency dynamics of the system. This may give useful information about the system in e.g. system identification. However, the neccessity of an anti-aliasing filter as described above removes this information if the sampling-frequency is low. Hence, to gain as much information about the system as possible it is preferable to sample very fast; at least so fast that the system is well damped above the Nyquist-frequency.

It should be noted, as illustrated in (Ljung 1987), that it is much less critical to sample too fast rather than too slow in estimation applications. The variance of parameter estimates in identification increases rapidly when the sampling rate is reduced below the time-constants of the considered system.

This is caused by lack of information about the system dynamics in the sampled output. Furthermore, if the noise characteristic is bandwidth limited, i.e. of low-pass nature, then no increase in the variance of the sampled signal is experienced when the sampling rate is increased if the sampling frequency is higher than twice the bandwidth of the noise.

Problems with fast sampling

An obvious problem with fast sampling is the cost of data acquisition. Clearly, if there is a cost involved in measuring the system output this puts a practical limit on the allowable rate of sampling. In digital control systems the required performance of the electronic equipment and thereby its cost, is increased with fast sampling rates. However, the rapid decrease in the price of microcomputers and electronic equipment in general makes this problem less important.

Another problem with fast sampling is the numerical difficulties that arise when using traditional shift-operator parametrizations. As we shall see later in Section 2.6 the poles of the sampled q-domain description will cluster around the point $(1, 0)$ in the complex plane. The problem is basically due to the fact that correlation between adjacent samples increases as the sampling rate is increased. This problem can be reduced by using alternative parametrizations which involve signal differencing rather than signal shifting. This is the subject of the next section.

2.5 δ-operator parametrizations

In this section we discuss an alternative parametrization of discrete-time systems, the so-called δ-operator parametrization. One of the most important works on such parametrizations is (Middleton and Goodwin 1990).

There the following points of motivation for this alternative discrete-time operator are given:

- It highlights similarities rather than differences between discrete-time and continuous-time systems. This allows physical continuous-time insights to be useful also in the discrete-time case.
- It allows a unified systems theory to be developed in which discrete-time and continuous-time results can be derived hand in hand rather than as two special cases.
- Most continuous-time results can be obtained as a limiting case (when the sampling period tends to zero) of the discrete-time results.
- It offers substantial numerical advantages especially at fast sampling rates.

We will illuminate these points in some detail in the following.

2.5.1 Definition

The delta-operator is defined by

$$\delta = \frac{q-1}{T} \tag{2.77}$$

or

$$q = 1 + T\delta \tag{2.78}$$

where T is the sampling period and q is the forward shift operator. Operating δ on a differentiable signal $x(t)$ gives

$$\delta x(t) = \frac{x(t+T) - x(t)}{T} \tag{2.79}$$

It is straightforward to see, that

$$\lim_{T \to 0} \delta x(t) = \frac{d}{dt} x(t) \tag{2.80}$$

which demonstrates the close relationship between the discrete-time δ-operator and the continuous-time differential operator at fast sampling rates. With this relationship in mind, $\delta^n x(t)$ will be called the n'th order δ-derivative of $x(t)$. This agrees with (Middleton and Goodwin 1990) where the term "generalized derivative" is used for d/dt in continuous-time and for δ in discrete-time.

Note, that (2.77) is a simple linear transformation and thus system modelling using δ-operator parametrizations offers exactly the same flexibility as q-operator parametrizations, i.e. the class of describable systems is not changed.

2.5.2 Stability region

The stability region of the δ-operator is obtained by a mapping of the stability region of the q-operator. From (2.77) it is clear, that the stability region of the δ-operator is enclosed by a circle with radius $1/T$ and center $(-1/T, 0)$. See also Figure 2.3.

Note, that the δ-domain stability region varies with the sampling period as opposed to the q-domain stability region. For $T \to 0$ the stability region converges to the open left-half plane, which coincides with the stability region for the continuous-time differential operator.

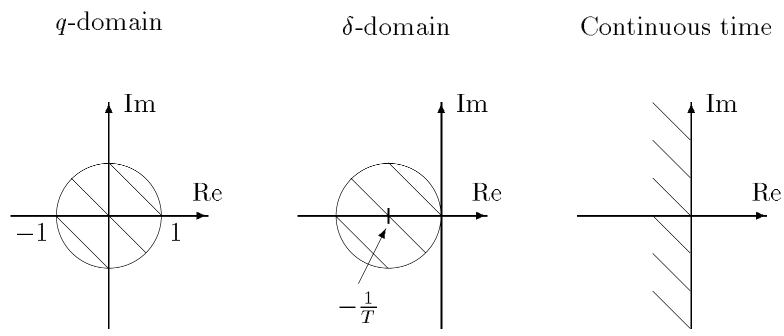

Figure 2.3: Stability regions for various operators.

2.5.3 Control design using the δ-operator

The use of the δ-operator in control theory is not new. The δ-operator is known by many as a tool for implementing continuous-time algorithms on digital computers. Consider Figure 2.4. The figure depicts a traditional procedure in designing digital controllers. One starts by modelling and identifying the controlled continuous-time process directly. A number of methods to identify continuous-time models are available, see (Unbehauen and Rao 1990). Typically, this involves some kind of approximation due to the fact, that only discrete-time measurements are available. One method is to transform the continuous-time model into a discrete-time approximation, for instance using the substitution $\frac{d}{dt} \to \delta$ or the Tustin approximation

$$\frac{d}{dt} \approx \frac{2}{T}\frac{q-1}{q+1} \tag{2.81}$$

Then one performs discrete-time estimation and transforms the result back into a continuous-time model estimate. Using this estimate a continuous-time controller is designed in the next block of Figure 2.4. Then the discrete-time nature of the process computer calls for a transformation of the continuous-time controller into an implementable digital form, again for instance using the δ-operator or the Tustin approximation. Two advantages of this procedure of design can be mentioned. Firstly, the use of continuous-time

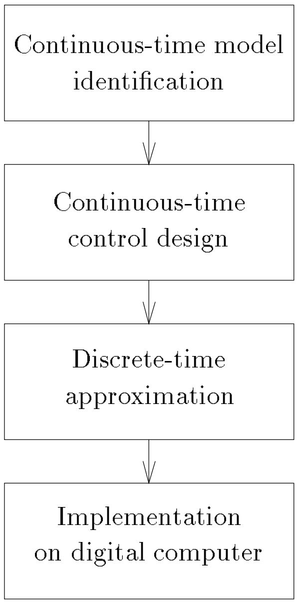

Figure 2.4: Designing digital control systems using continuous-time control theory and continuous-time identification.

models makes it possible to utilize a priori physical knowledge of the system under control, both in estimation and control design. Secondly, control design criteria are often more intuitively appealing in continuous-time as they can be directly related to physical properties of the controlled system. One disadvantage of the approach of Figure 2.4 is the involved approximations, especially in the actual control implementation. With optimal control designs the optimality is lost during implementation, that is, one does not achieve a minimization of the control criterion due to approximations in implementation.

Another approach is depicted in Figure 2.5. The use of digital computers makes it more natural to work directly with discrete-time models and

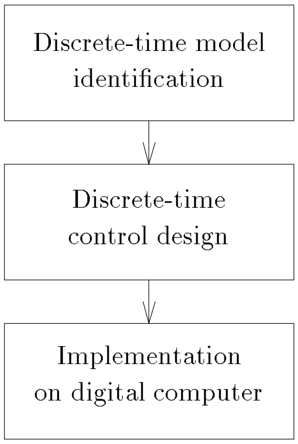

Figure 2.5: Designing digital control systems using discrete-time control theory and discrete-time identification.

control design algorithms. Based on sampled data a discrete-time model is estimated. Using this model a discrete-time control design is performed. The obtained controller can be directly implemented without approximation. The obvious disadvantage of this approach is that possible a priori physical knowledge of the continuous-time process can seemingly not be utilized in a simple way. This is where the δ-operator comes into the picture. If we view the δ-operator as a convenient discrete-time operator it is possible to estimate directly a discrete-time model based on sampled data and implement exactly the obtained discrete-time control design. However, as discussed later, the close correspondence to continuous-time of δ-operator models allows the use of physical a priori knowledge and design of discrete-time control algorithms which approximate well continuous-time designs. This is the use of δ-operator parametrizations that is considered in this thesis.

2.5.4 State-space representations

For implementational reasons it is interesting to investigate the connection between state space descriptions in the δ-domain and the q-domain. Consider the stochastic state space description

$$\begin{aligned} \delta x(t) &= \boldsymbol{A}_\delta x(t) + \boldsymbol{B}_\delta u(t) + v_\delta(t) \\ y(t) &= \boldsymbol{C}_\delta x(t) + e_\delta(t) \end{aligned} \tag{2.82}$$

where $v_\delta(t)$ and $e_\delta(t)$ is state excitation noise and measurement noise, respectively. The two noise sources are assumed to be mutually independent discrete-time white noise processes. Using (2.77) the state update can be written as

$$\begin{aligned} qx(t) &= x(t) + T\boldsymbol{A}_\delta x(t) + T\boldsymbol{B}_\delta u(t) + Tv_\delta(t) && (2.83) \\ &= (\boldsymbol{I} + T\boldsymbol{A}_\delta)x(t) + T\boldsymbol{B}_\delta u(t) + Tv_\delta(t) && (2.84) \end{aligned}$$

In the q-domain the corresponding stochastic state space description is (2.49)

$$\begin{aligned} qx(t) &= \boldsymbol{A}_q x(t) + \boldsymbol{B}_q u(t) + v(t) \\ y(t) &= \boldsymbol{C}_q x(t) + e(t) \end{aligned} \tag{2.85}$$

We see, that in order to get equivalent models in the two domains we must select

$$\begin{aligned} v_\delta(t) &= \frac{1}{T} v(t) && (2.86) \\ e_\delta(t) &= e(t) && (2.87) \end{aligned}$$

Hence, the noise variances are

$$\begin{aligned} \boldsymbol{Q}_\delta &= \mathcal{E}\{v_\delta(t) v_\delta(t)^\top\} = \frac{1}{T^2}\boldsymbol{Q}_q && (2.88) \\ \boldsymbol{R}_\delta &= \mathcal{E}\{e_\delta(t) e_\delta(t)^\top\} = \boldsymbol{R}_q && (2.89) \end{aligned}$$

As noted in (Jazwinski 1970) a continuous-time white-noise process may be regarded as the limit as the sampling period tends to zero of a discrete-time white noise sequence provided, that the variance of the discrete-time noise is chosen as the fraction between the continuouse-time white noise intensity and the sampling rate. To keep the connection to the continuous-time case we therefore define the δ-domain intensities (discrete-time spectral densities obtained by dividing the variances with the sampling frequency $1/T$)

$$\begin{aligned} \boldsymbol{\Omega}_\delta &= T\boldsymbol{Q}_\delta = \frac{1}{T}\boldsymbol{Q}_q && (2.90) \\ \boldsymbol{\Gamma}_\delta &= T\boldsymbol{R}_\delta = T\boldsymbol{R}_q && (2.91) \end{aligned}$$

The connection between the state-space matrices in the q-domain and δ-domain follows directly by inspection

$$\begin{aligned} \boldsymbol{A}_q &= \boldsymbol{I} + T\boldsymbol{A}_\delta && (2.92) \\ \boldsymbol{B}_q &= T\boldsymbol{B}_\delta && (2.93) \\ \boldsymbol{C}_q &= \boldsymbol{C}_\delta && (2.94) \end{aligned}$$

The above equations provide a direct connection between the state-space matrices of the q-domain and the δ-domain. This could lead one to calculate the δ-domain matrices as follows: Using existing routines calculate the q-domain representation. Then transform into the δ-domain by

$$\begin{aligned} \boldsymbol{A}_\delta &= \frac{\boldsymbol{A}_q - \boldsymbol{I}}{T} && (2.95) \\ \boldsymbol{B}_\delta &= \frac{\boldsymbol{B}_q}{T} && (2.96) \end{aligned}$$

Although this is mathematically correct it is not advisable to determine the δ-domain realization in this way as the poor numerical properties of the q-domain representation, especially at fast sampling rates, is then carried over to the δ-domain. A much better procedure is to directly derive the δ-domain matrices as suggested in (Middleton and Goodwin 1990) by

$$\begin{aligned} \boldsymbol{A}_\delta &= \boldsymbol{X}\boldsymbol{A}_c && (2.97) \\ \boldsymbol{B}_\delta &= \boldsymbol{X}\boldsymbol{B}_c && (2.98) \end{aligned}$$

where $\boldsymbol{A}_c$ and $\boldsymbol{B}_c$ are the continuous-time matrices and

$$\boldsymbol{X} = \frac{1}{T}\int_0^T e^{\boldsymbol{A}_c\tau}\,d\tau = \boldsymbol{I} + \frac{\boldsymbol{A}_c T}{2!} + \frac{\boldsymbol{A}_c^2 T^2}{3!} + \cdots \qquad (2.99)$$

Note, that $\lim_{T\to 0} \boldsymbol{X} = \boldsymbol{I}$ which implies

$$\lim_{T\to 0} \boldsymbol{A}_\delta = \boldsymbol{A}_c \tag{2.100}$$

$$\lim_{T\to 0} \boldsymbol{B}_\delta = \boldsymbol{B}_c \tag{2.101}$$

This shows the close connection between the δ-domain form and the continuous-time model as this is recovered in the limit case. Further information on computation of δ-domain representations can be found in (Neumann 1993).

2.5.5 δ-form of discrete-time optimal filtering

The δ-form of the discrete-time optimal filter can be obtained easily from the q-domain form presented in Theorem 3.

Theorem 5 *(Middleton and Goodwin 1990)*

Consider the sampled system description

$$\delta x(t) = \boldsymbol{A}_\delta x(t) + \boldsymbol{B}_\delta u(t) + v_\delta(t) \tag{2.102}$$

$$y(t) = \boldsymbol{C}_\delta x(t) + e_\delta(t) \qquad t = kT,\ k \geq k_0 \tag{2.103}$$

with initial state $x(k_0T) \in N(m_0, \boldsymbol{P}_0)$ independent of $v_\delta(t)$ and $e_\delta(t)$. The noise sequences $v_\delta(t)$ and $e_\delta(t)$ are mutually independent discrete-time white noise processes with variance $\boldsymbol{Q}_\delta = \boldsymbol{\Omega}_\delta/T$ and $\boldsymbol{R}_\delta = \boldsymbol{\Gamma}_\delta/T > 0$, respectively.

The discrete-time optimal observer which minimizes for all $t = kT,\ k \geq k_0$

$$\mathcal{E}\left\{[x(t) - \hat{x}(t)]^\top [x(t) - \hat{x}(t)]\right\} \tag{2.104}$$

is given by

$$\begin{aligned} \delta\hat{x}(t) &= \boldsymbol{A}_\delta \hat{x}(t) + \boldsymbol{B}_\delta u(t) + \boldsymbol{K}_\delta(t)\tilde{y}(t) \\ \tilde{y}(t) &= y(t) - \boldsymbol{C}_\delta \hat{x}(t) \end{aligned} \tag{2.105}$$

where the gain matrix $\boldsymbol{K}_\delta(t)$ is chosen as

$$\boldsymbol{K}_\delta(t) = (T\boldsymbol{A}_\delta + \boldsymbol{I})\boldsymbol{P}_\delta(t)\boldsymbol{C}_\delta^\top \left(T\boldsymbol{C}_\delta \boldsymbol{P}_\delta(t)\boldsymbol{C}_\delta^\top + \boldsymbol{\Gamma}_\delta\right)^{-1} \tag{2.106}$$

with $\boldsymbol{P}_\delta(t)$ being the solution to the δ-domain matrix Riccati equation

$$\begin{aligned} \delta\boldsymbol{P}_\delta(t) &= \boldsymbol{A}_\delta \boldsymbol{P}_\delta(t) + \boldsymbol{P}_\delta(t)\boldsymbol{A}_\delta^\top + \boldsymbol{\Omega}_\delta + T\boldsymbol{A}_\delta \boldsymbol{P}_\delta(t)\boldsymbol{A}_\delta^\top \\ &\quad - \boldsymbol{K}_\delta(t)\left(T\boldsymbol{C}_\delta \boldsymbol{P}_\delta(t)\boldsymbol{C}_\delta^\top + \boldsymbol{\Gamma}_\delta\right)\boldsymbol{K}_\delta(t)^\top \end{aligned} \tag{2.107}$$

with initial conditions

$$\boldsymbol{P}_\delta(k_0T) = \boldsymbol{P}_0 \tag{2.108}$$

$$\hat{x}(k_0T) = m_0 \tag{2.109}$$

The variance on the error of the state estimate is

$$\mathcal{E}\left\{[x(t)-\hat{x}(t)][x(t)-\hat{x}(t)]^\top\right\} = \boldsymbol{P}_\delta(t), \qquad t = kT \tag{2.110}$$

□

Note, that we have used the shorthand notation

$$\delta\hat{x}(t) = \frac{\hat{x}(t+T|t) - \hat{x}(t|t-T)}{T} \tag{2.111}$$

with the definition of the state-estimate given in (2.57).

2.5.6 Transfer function representations

As in the q-domain case we must find the stationary solution to the optimal filtering problem in order to derive the innovations form which leads us to the transfer function representation.

Theorem 6 *(Middleton and Goodwin 1990)*

Consider the optimal discrete-time filter of Theorem 5. Factor the state-noise intensity as

$$\boldsymbol{\Omega}_\delta = \boldsymbol{\Omega}_\delta^{1/2}(\boldsymbol{\Omega}_\delta^{1/2})^\top \tag{2.112}$$

Then if $(\boldsymbol{C}_\delta, \boldsymbol{A}_\delta)$ is detectable and $(\boldsymbol{A}_\delta, \boldsymbol{\Omega}_\delta^{1/2})$ is reachable the stationary innovations representation exists and is given by

$$\begin{aligned} \delta x(t) &= \boldsymbol{A}_\delta x(t) + \boldsymbol{B}_\delta u(t) + \boldsymbol{K}_\delta \tilde{y}(t) && (2.113)\\ y(t) &= \boldsymbol{C}_\delta x(t) + \tilde{y}(t) && (2.114) \end{aligned}$$

where $\boldsymbol{P}_\delta$ is the unique positive definite solution to the discrete-time δ-domain algebraic Riccati equation

$$\begin{aligned} 0 = {} & \boldsymbol{A}_\delta \boldsymbol{P}_\delta + \boldsymbol{P}_\delta \boldsymbol{A}_\delta^\top + \boldsymbol{\Omega}_\delta + T\boldsymbol{A}_\delta \boldsymbol{P}_\delta \boldsymbol{A}_\delta^\top \\ & - \boldsymbol{K}_\delta \left(T\boldsymbol{C}_\delta \boldsymbol{P}_\delta \boldsymbol{C}_\delta^\top + \boldsymbol{\Gamma}_\delta \right) \boldsymbol{K}_\delta^\top \end{aligned} \tag{2.115}$$

and the Kalman gain is given as

$$\boldsymbol{K}_\delta = (T\boldsymbol{A}_\delta + \boldsymbol{I})\boldsymbol{P}_\delta \boldsymbol{C}_\delta^\top \left(T\boldsymbol{C}_\delta \boldsymbol{P}_\delta \boldsymbol{C}_\delta^\top + \boldsymbol{\Gamma}_\delta \right)^{-1} \tag{2.116}$$

The innovation process $\tilde{y}(t)$ is a white noise sequence with variance

$$\mathcal{E}\{\tilde{y}(t)\tilde{y}(t)^\top\} = \boldsymbol{C}_\delta \boldsymbol{P}_\delta \boldsymbol{C}_\delta^\top + \frac{\boldsymbol{\Omega}_\delta}{T} \tag{2.117}$$

□

The algebraic Riccati equation (2.115) is the stationary solution to the δ-domain Riccati difference equation (2.107) for $t_0 \to -\infty$. In stationarity it holds, that $\delta P_\delta(t) = 0$.

The stochastic state space description (2.82) can now be transformed to a external δ-domain ARMAX description using rewritings similar to those in the q-domain case.

$$y(t) = \boldsymbol{C}_\delta \left(\delta \boldsymbol{I} - \boldsymbol{A}_\delta\right)^{-1} \left[\boldsymbol{B}_\delta u(t) + \boldsymbol{K}_\delta \tilde{y}(t)\right] + \tilde{y}(t) \tag{2.118}$$

which gives

$$\begin{aligned}\det(\delta \boldsymbol{I} - \boldsymbol{A}_\delta) y(t) &= \boldsymbol{C}_\delta \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A}_\delta) \boldsymbol{B}_\delta u(t) \\ &\quad + (\boldsymbol{C}_\delta \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A}_\delta) \boldsymbol{K}_\delta + \det(\delta \boldsymbol{I} - \boldsymbol{A}_\delta))\, \tilde{y}(t)\end{aligned}$$

From this we identify the ARMAX description

$$A_\delta(\delta) y(t) = B_\delta(\delta) u(t) + C_\delta(\delta) \tilde{y}(t) \tag{2.119}$$

with

$$\begin{aligned}A_\delta(\delta) &= \det(\delta \boldsymbol{I} - \boldsymbol{A}_\delta) &(2.120)\\ B_\delta(\delta) &= \boldsymbol{C}_\delta \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A}_\delta) \boldsymbol{B}_\delta &(2.121)\\ C_\delta(\delta) &= \boldsymbol{C}_\delta \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A}_\delta) \boldsymbol{K}_\delta + \det(\delta \boldsymbol{I} - \boldsymbol{A}_\delta) &(2.122)\end{aligned}$$

where in general

$$\begin{aligned}\deg(A_\delta) &= n &(2.123)\\ \deg(B_\delta) &= m \leq n-1 &(2.124)\\ \deg(C_\delta) &= n &(2.125)\end{aligned}$$

We note, that $A_\delta(\delta)$ is monic.

Connection to q-domain description

The q-domain ARMAX description (2.66) and the δ-domain version (2.119) are derived from equivalent sampled descriptions and are consequently input-output equivalent. Hence, a direct correspondence exists between the system polynomials in the two domains. By using the definition of the δ-operator it is not difficult to see, that

$$\begin{aligned}A_\delta(\delta) &= \frac{1}{T^n} A_q(1 + T\delta) &(2.126)\\ B_\delta(\delta) &= \frac{1}{T^n} B_q(1 + T\delta) &(2.127)\\ C_\delta(\delta) &= \frac{1}{T^n} C_q(1 + T\delta) &(2.128)\end{aligned}$$

where the factor $1/T^n$ is necessary to keep $A_\delta(\delta)$ monic.

The above relations may be used to derive explicit expressions for the connection between the parameters of the polynomials in the two domains. Define the parameters of the δ-domain polynomials by

$$\begin{aligned}
A_\delta(\delta) &= \delta^n + a_{\delta,n-1}\delta^{n-1} + \cdots + a_{\delta,0} && (2.129)\\
B_\delta(\delta) &= b_{\delta,m}\delta^m + b_{\delta,m-1}\delta^{m-1} + \cdots + b_{\delta,0} && (2.130)\\
C_\delta(\delta) &= \delta^n + c_{\delta,n-1}\delta^{n-1} + \cdots + c_{\delta,0} && (2.131)
\end{aligned}$$

and similarly in the q-domain

$$\begin{aligned}
A_q(q) &= q^n + a_{q,n-1}q^{n-1} + \cdots + a_{q,0} && (2.132)\\
B_q(q) &= b_{q,m}q^m + b_{q,m-1}q^{m-1} + \cdots + b_{q,0} && (2.133)\\
C_q(q) &= q^n + c_{q,n-1}q^{n-1} + \cdots + c_{q,0} && (2.134)
\end{aligned}$$

where the degree of $B_\delta(\delta)$ and $B_q(q)$ is assumed to be $m < n$ so that the time delay in samples is $d = n - m$.

Consider the rewritings

$$\begin{aligned}
A_\delta(\delta) &= \frac{1}{T^n}\sum_{i=0}^{n} a_{q,i}q^i && (2.135)\\
&= \frac{1}{T^n}\sum_{i=0}^{n} a_{q,i}(1+\delta T)^i && (2.136)\\
&= \frac{1}{T^n}\sum_{i=0}^{n}\sum_{k=0}^{i} a_{q,i}\binom{i}{k}T^k\delta^k && (2.137)\\
&= \sum_{j=0}^{n} a_{\delta,j}\delta^j && (2.138)
\end{aligned}$$

where we have used the binomial formula

$$(1+\delta T)^i = \sum_{k=0}^{i}\binom{i}{k}(T\delta)^i \qquad (2.139)$$

and the binomial coefficients are defined by

$$\binom{i}{k} = \frac{i!}{k!(i-k)!} \tag{2.140}$$

By comparison of coefficients it now appears, that

$$\boxed{a_{\delta,j} = \sum_{i=j}^{n} a_{q,i} \binom{i}{j} T^{j-n}, \qquad j = 0, \ldots, n \qquad (2.141)}$$

On noting that

$$\begin{aligned}
A_q(q) &= T^n A_\delta(\frac{q-1}{T}) && (2.142)\\
&= T^n \sum_{i=0}^{n} a_{\delta,i} \left(\frac{q-1}{T}\right)^i && (2.143)\\
&= \sum_{i=0}^{n} T^{n-i} a_{\delta,i} \sum_{k=0}^{i} \binom{i}{k} (-1)^{k-i} q^k && (2.144)
\end{aligned}$$

and again comparing coefficients one finds the inverse relation

$$\boxed{a_{q,j} = \sum_{i=j}^{n} T^{n-i} a_{\delta,i} \binom{i}{j} (-1)^{j-i}, \qquad j = 0, \ldots, n \qquad (2.145)}$$

Note that due to the monicness of $A_q(q)$ and $A_\delta(\delta)$

$$a_{q,n} = a_{\delta,n} = 1 \tag{2.146}$$

In vector form these conversions can be expressed as

$$\boldsymbol{a}_q = \boldsymbol{M}_{q\delta}(n)\boldsymbol{a}_\delta \tag{2.147}$$

where

$$\boldsymbol{a}_q = [a_{q,0} \ a_{q,1} \ \ldots \ a_{q,n-1} \ 1]^\top \tag{2.148}$$

$$\boldsymbol{a}_\delta \quad = \quad [a_{\delta,0}\ a_{\delta,1}\ \ldots\ a_{\delta,n-1}\ 1]^\top \tag{2.149}$$

and where the conversion matrix is upper triangular and given by

$$\boldsymbol{M}_{q\delta}(k) = \begin{bmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,k} \\ 0 & m_{1,1} & \cdots & m_{1,k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & m_{k,k} \end{bmatrix} \tag{2.150}$$

with

$$m_{j,i} = \begin{cases} (-1)^{j-i} T^{k-i} \binom{i}{j} & \text{for } i \geq j \\ 0 & \text{otherwise} \end{cases} \tag{2.151}$$

Similarly,

$$\boldsymbol{a}_\delta = \boldsymbol{M}_{\delta q}(n) \boldsymbol{a}_q \tag{2.152}$$

with

$$\boldsymbol{M}_{\delta q}(k) = \boldsymbol{M}_{q\delta}^{-1}(k) = \begin{bmatrix} m'_{0,0} & m'_{0,1} & \cdots & m'_{0,k} \\ 0 & m'_{1,1} & \cdots & m'_{1,k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & m'_{k,k} \end{bmatrix} \tag{2.153}$$

where the matrix entries are

$$m'_{j,i} = \begin{cases} T^{j-k} \binom{i}{j} & \text{for } i \geq j \\ 0 & \text{otherwise} \end{cases} \tag{2.154}$$

Finally, with

$$\boldsymbol{b}_q \quad = \quad [b_{q,0}\ b_{q,1}\ \ldots\ b_{q,m}]^\top \tag{2.155}$$

$$\boldsymbol{b}_\delta \quad = \quad [b_{\delta,0}\ b_{\delta,1}\ \ldots\ b_{\delta,m}]^\top \tag{2.156}$$

$$\boldsymbol{c}_q \quad = \quad [c_{q,0}\ c_{q,1}\ \ldots\ c_{q,n}]^\top \tag{2.157}$$

$$\boldsymbol{c}_\delta \quad = \quad [c_{\delta,0}\ c_{\delta,1}\ \ldots\ c_{\delta,n}]^\top \tag{2.158}$$

the following can easily be verified

$$\begin{aligned}
\boldsymbol{b}_q &= T^{n-m}\boldsymbol{M}_{q\delta}(m)\boldsymbol{b}_\delta && (2.159)\\
\boldsymbol{b}_\delta &= T^{m-n}\boldsymbol{M}_{\delta q}(m)\boldsymbol{b}_q && (2.160)\\
\boldsymbol{c}_q &= \boldsymbol{M}_{q\delta}(n)\boldsymbol{c}_\delta && (2.161)\\
\boldsymbol{c}_\delta &= \boldsymbol{M}_{\delta q}(n)\boldsymbol{c}_q && (2.162)
\end{aligned}$$

We have now established simple explicit conversion formulas for the parameters of the transfer function descriptions in the two domains.

2.6 Limiting properties of discrete-time models

In this section we will investigate more closely the limiting properties as the sampling period tends to zero of the different discrete-time models, that have been introduced. We consider state-space representations and transfer function representations separately.

2.6.1 State-space representations

Consider again the stochastic continuous-time state-space model

$$\begin{aligned}
\dot{x}_c(t) &= \boldsymbol{A}_c x_c(t) + \boldsymbol{B}_c u(t) + \dot{v}_c(t)\\
y_c(t) &= \boldsymbol{C}_c x_c(t) + \dot{e}_c(t) && (2.163)
\end{aligned}$$

where the noise sequences $\dot{v}_c(t)$ and $\dot{e}_c(t)$ are mutually independent zero-mean white noise with constant intensities $\boldsymbol{\Omega}_c$ and $\boldsymbol{\Gamma}_c > 0$, respectively. Below we summarize the limiting properties of the discrete-time q-domain and δ-domain models.

q-domain models

The limit values of the system matrices of the q-domain state-space representation follow from (2.38), (2.39) and (2.46). It is easy to see, that

$$\lim_{T\to 0} \boldsymbol{A}_q = \boldsymbol{I} \tag{2.164}$$
$$\lim_{T\to 0} \boldsymbol{B}_q = \boldsymbol{0} \tag{2.165}$$
$$\lim_{T\to 0} \boldsymbol{C}_q = \boldsymbol{C}_c \tag{2.166}$$

This is totally misleading and points out, that shift-operator models are unusable at fast sampling rates. The problem is, that with a short sampling period there will be almost no change in successive samples of state, that is $x(t+T) \approx x(t)$. Also the effect of the input from one sample to the next will cease at fast sampling. These observations are reflected by the limit results given above. In addition the noise-variances are misleading at fast sampling. The effect of the state noise is ignorable similarly to the system input. This is evident from (2.43) which gives

$$\lim_{T\to 0} \boldsymbol{Q}_q = \boldsymbol{0} \tag{2.167}$$

As discussed earlier the variance of the measurement noise will grow approximately linearly with $1/T$ as $T \to 0$, see (2.47), i.e.

$$\lim_{T\to 0} \boldsymbol{R}_q = \infty \tag{2.168}$$

It turns out, that it is more appropriate to consider spectral densities as we have done for the δ-domain descriptions. As opposed to the noise variances, these spectral densities are roughly invariant as the sampling rate changes. Also the Kalman gain for the discrete-time Kalman predictor is misleading in the limit case, as (2.168) and (2.52) lead to

$$\lim_{T\to 0} \boldsymbol{K}_q(t) = 0 \tag{2.169}$$

δ-domain models

The δ-domain descriptions result in more reasonable limiting properties of the discrete-time description.

We have already seen in (2.100), (2.101) and (2.94) combined with (2.166), that

$$\lim_{T\to 0} \boldsymbol{A}_\delta = \boldsymbol{A}_c \tag{2.170}$$

$$\lim_{T\to 0} \boldsymbol{B}_\delta = \boldsymbol{B}_c \tag{2.171}$$

$$\lim_{T\to 0} \boldsymbol{C}_\delta = \boldsymbol{C}_c \tag{2.172}$$

That is, the continuous-time state-space system matrices are recovered in the limit case. In the δ-domain we introduced discrete-time spectral densities to describe the noise processes. These converge to the corresponding continuous-time spectral densities

$$\lim_{T\to 0} \boldsymbol{\Omega}_\delta = \boldsymbol{\Omega}_c \tag{2.173}$$

$$\lim_{T\to 0} \boldsymbol{\Gamma}_\delta = \boldsymbol{\Gamma}_c \tag{2.174}$$

which follows from (2.47) and (2.43) combined with (2.90) and (2.91). It appears, that at fast sampling rates these δ-domain spectral intensities are roughly invariant.

Consider the δ-domain Kalman filter given in Theorem 5. It is easy to see, using the above and letting $T \to 0$, that the δ-domain Kalman gain has the following limiting property

$$\lim_{T\to 0} \boldsymbol{K}_\delta(t) = \boldsymbol{P}_\delta(t)\boldsymbol{C}_c^\top \boldsymbol{\Gamma}_c^{-1} \tag{2.175}$$

where the mean-square of the reconstruction error follows the differential equation

$$\dot{\boldsymbol{P}}_\delta(t) = \boldsymbol{A}_c\boldsymbol{P}_\delta(t) + \boldsymbol{P}_\delta(t)\boldsymbol{A}_c^\top + \boldsymbol{\Omega}_c - \boldsymbol{K}_\delta(t)\boldsymbol{\Gamma}_c\boldsymbol{K}_\delta(t)^\top \tag{2.176}$$

where we have utilized the convergence of the δ-operator to the differential operator. Comparing with the continuous-time filter equations (2.26) and (2.27) we observe an exact correspondence and hence, we conclude that with compatible initial conditions

$$\lim_{T\to 0} \boldsymbol{K}_\delta(t) = \boldsymbol{K}_c(t) \tag{2.177}$$

$$\lim_{T\to 0} \boldsymbol{P}_\delta(t) = \boldsymbol{P}_c(t) \tag{2.178}$$

These limiting properties show that a unified theory for continuous and discrete-time optimal filtering is possible. In fact the convergence as $T \to 0$ of δ-operator models to the continuous-time equivalent enables an almost complete unified theory for discrete and continuous-time as pursued in (Middleton and Goodwin 1990).

Also the corresponding spectral density for the variance of the innovation process (2.117) converges to the continuous-time spectral density given in (2.36)

$$\lim_{T\to 0} T\left(\boldsymbol{C}_\delta \boldsymbol{P}_\delta \boldsymbol{C}_\delta^\top + \frac{\boldsymbol{\Omega}_\delta}{T}\right) = \boldsymbol{\Omega}_c \tag{2.179}$$

where (2.173) has been used. This of course also holds for the q-domain representation as the innovation process is identical in the two cases.

Not surprisingly, these nice limit results for δ-domain state-space representations carries over to the transfer function forms as investigated next.

2.6.2 Transfer function representations

Consider a deterministic continuous-time system with transfer function ($m < n$)

$$G(s) = \frac{B'(s)}{A'(s)} = \frac{b'_m s^m + b'_{m-1} s^{m-1} + \cdots + b'_0}{s^n + a'_{n-1} s^{n-1} + \cdots + a'_0} \tag{2.180}$$

where s is the complex Laplace-operator and assume that a discrete-time transfer function

$$G_\delta(\delta) = \frac{B_\delta(\delta)}{A_\delta(\delta)} \tag{2.181}$$

is obtained by sampling of $G(s)$ with a zero-order hold circuit and sampling time T. This will usually give rise to a $B_\delta(\delta)$ polynomial of order $n-1$ and an $A_\delta(\delta)$ polynomial of order n. It has been shown in e.g. (Goodwin *et al.* 1986), that when $T \to 0$ then

$$\begin{aligned} a_{\delta,i} &\to a'_i && i = 1, \ldots, n \\ b_{\delta,i} &\to b'_i && i = 0, \ldots, m \\ b_{\delta,i} &\to 0 && i = m+1, \ldots, n-1 \end{aligned} \tag{2.182}$$

This emphasizes the relationship between continuous-time descriptions and δ-operator descriptions. For small sampling intervals the continuous-time parameters can be recovered in the discrete-time δ description. This allows the inclusion of (physical) a priori knowledge in e.g. parameter estimation.

A similar simple relationship between the transfer function parameters in continuous and discrete time does not exist for descriptions in the q-domain. In particular the effect of the sampling process will be scattered on all parameters in $A_q(q)$ and $B_q(q)$. This makes it virtually impossible to utilize physical knowledge of the continuous-time transfer function parameters.

2.6.3 Poles and zeros of sampled transfer functions

Let us now consider the pole and zero locations of the sampled transfer function. Consider the continuous-time system

$$G(s) = \frac{B'(s)}{A'(s)} = K \frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)} \tag{2.183}$$

where z_i and p_i denotes the zeros and poles of the transfer function, respectively, and $n > m$. The system is now sampled directly with zero order hold on the input to give both δ-domain and the q-domain descriptions. The results given in the following can be found in (Middleton and Goodwin 1990, Goodwin *et al.* 1986)

Pole locations

In the q-domain the poles of the sampled transfer function are obtained by a mapping of the continuous-time poles

$$p_{q,i} = e^{p_i T} , \qquad i = 1, \ldots, n \tag{2.184}$$

We see, that at fast sampling rates

$$\lim_{T \to 0} p_{q,i} = 1 , \qquad i = 1, \ldots, n \tag{2.185}$$

That is, the poles of the q-domain representation cluster around the point $(1, 0)$ in the complex plane. It seems reasonable, that a much better numerical precision could be obtained by shifting the origin from $(1, 0)$ to $(0, 0)$ in the complex plane. This is exactly what is achieved by using the δ-operator. In the δ-domain the poles become

$$p_{\delta,i} = \frac{e^{p_i T} - 1}{T} , \qquad i = 1, \ldots, n \tag{2.186}$$

and the continuous-time poles are recovered in the limit case

$$\lim_{T \to 0} p_{\delta,i} = p_i , \qquad i = 1, \ldots, n \tag{2.187}$$

Zero locations

The zero locations of the sampled system can in general not be determined by a simple mapping of the continuous-time zeros. In addition some extra zeros are introduced by the sampling process if the relative degree of the continuous-time transfer function is greater than 1. This follows from the fact, that the orders of $B_\delta(\delta)$ and $B_q(q)$ in general are $n-1$, while the order of $B'(s)$ is m. It is possible to divide the zeros of the sampled system into two categories; zeros that are due to the continuous-time zeros and zeros introduced by the sampling process. The latter category is referred to as sampling zeros.

Some limit results for small sampling periods were found in (Åström *et al.* 1984). The limiting behaviour of sampled transfer functions has been thoroughly investigated in (Wahlberg 1987, Thygesen 1993), the latter also for systems with time delays, that are not a multiple of the sampling period. Some of the results are now repeated here. Other work on zeros in sampled transfer functions can be found in (Hagiwara 1996) which studies the mapping of the continuous-time zeros, (Ishitobi 1992) which derives conditions for the stability of the discrete-time zeros and (Tesfaye and Tomizuka 1995) where the neglectability of the zeros of δ-operator parametrizations at fast sampling rates is studied.

In the q-domain we may write as $T \to 0$

$$B_q(q) = K(q-1)^m B_{q,n-m}(q) \tag{2.188}$$

where $B_{q,n-m}(q)$ is a polynomial of degree $n-m-1$. The sampling zeros can be found as the zeros of $B_{q,n-m}(q)$. Note, that these zeros depend only on the relative degree of the continuous-time transfer function. As shown in (Åström *et al.* 1984) the zeros of $B_{q,n-m}$ are located at fixed locations in the left half plane. At least one zero of $B_{q,n-m}(q)$ will be non-minimum phase (strictly outside the unit circle) if the relative degree of the continuous-time system is greater than 2, i.e. $n-m>2$. Hence, with sufficiently fast sampling the sampled transfer function will be non-minimum phase if $n-m>2$ even if the continuous-time system is minimum-phase. Many control designs will then fail, e.g. minimum-variance (MV0) control. This is a highly undesirable discrepancy between the continuous-time and the sampled system descriptions. The remaining m zeros, which can be thought of as mapped continuous time zeros, will tend to 1 as $e^{z_i T}$.

The results in the δ-domain can be obtained by transforming the q-domain results. It can be shown, that with

$$B_\delta(\delta) = K \prod_{i=1}^{n-1} (\delta - z_{\delta,i}) \tag{2.189}$$

then

$$\lim_{T \to 0} z_{\delta,i} = z_i \,, \qquad i = 1, \ldots, m \tag{2.190}$$

and

$$\lim_{T \to 0} \prod_{i=m+1}^{n-1} (\delta - z_{\delta,i}) = B_{\delta,n-m}(\delta) \tag{2.191}$$

where $B_{\delta,n-m} = B_{q,n-m}(1+\delta T)$ is a polynomial of degree $n-m-1$. Again the limiting sampling zeros are found as the zeros of a fixed polynomial $B_{\delta,n-m}(\delta)$ and at least one zero is unstable if $n - m > 2$. Moreover

$$\lim_{T \to 0} z_{\delta,i} = -\infty \,, \qquad i = m+1, \ldots, n-1 \tag{2.192}$$

The mapped continuous-time zeros are recovered in the discrete description as is evident from (2.190).

From (2.182) it appears, that the sampling zeros can be associated with the highest order terms of the numerator polynomial $B_\delta(\delta)$. As mentioned this opens a possibility of discarding these high order terms in the control design. In that way, control designs will not fail because of non-minimum phase zeros introduced by the sampling process. An example of such an approach is found in (Goodwin *et al.* 1986). A more detailed analysis of this property of δ-domain parametrizations can be found in (Tesfaye and Tomizuka 1995).

2.7 Parameter estimation in δ-operator parametrizations

As investigated in (Gevers and Li 1993), reparametrization of a transfer function to be estimated can give better numerical properties and a faster convergence of the parameter estimates. In this section we will examine briefly how estimation of transfer functions in the δ-domain can be performed and give some connections to q-domain estimation.

Parametrizations in the δ-operator may also be useful in identification of state-space models. An example on this can be found in (Rostgaard *et al.* 1994b) which considers maximum likelihood estimation of a flexible robot model. Subspace identification of δ-operator state-space models is considered in (Katayama *et al.* 1995). In (Tesfaye and Tomizuka 1996) the effects of system perturbations in the design of a robust digital tracking controller are analyzed using δ-domain state-space models at fast sampling rates.

2.7.1 δ-domain ARX models

Consider the δ-domain ARX model

$$A_\delta(\delta)y(t) = B_\delta(\delta)u(t) + e(t) \tag{2.193}$$

where $\{e(t)\}$ is a white noise sequence and

$$\begin{aligned} A_\delta(\delta) &= \delta^n + a_{\delta,n-1}\delta^{n-1} + \cdots + a_{\delta,0} &(2.194)\\ B_\delta(\delta) &= b_{\delta,m}\delta^m + b_{\delta,m-1}\delta^{m-1} + \cdots + b_{\delta,0} &(2.195)\end{aligned}$$

with $m < n$ to ensure a causal system with a time-delay of at least one sample. The input-output relation can equivalently be written as

$$\delta^n y(t) = -\sum_{i=0}^{n-1} a_{\delta,i}\delta^i y(t) + \sum_{i=0}^{m} b_{\delta,i}\delta^i u(t) + e(t) \tag{2.196}$$

The signals appearing in the above expression are future signals since the δ-operator is non-causal. To obtain a relation with known signals we apply an at least n'th order stable filter

$$\frac{1}{F(\delta)} = \frac{1}{\delta^n + f_{n-1}\delta^{n-1} + \cdots + f_0} \tag{2.197}$$

We then get

$$\delta^n y^f(t) = -\sum_{i=0}^{n-1} a_{\delta,i}\delta^i y^f(t) + \sum_{i=0}^{m} b_{\delta,i}\delta^i u^f(t) + e^f(t) \tag{2.198}$$

where

$$y^f(t) = \frac{y(t)}{F(\delta)}, \qquad u^f(t) = \frac{u(t)}{F(\delta)}, \qquad e^f(t) = \frac{e(t)}{F(\delta)} \tag{2.199}$$

The filtering operation can be perceived as an estimation of the regression signals. In regression form (2.198) can be written as

$$\delta^n y^f(t) = \varphi(t)^T\theta_0 + e^f(t) \tag{2.200}$$

where

$$\varphi = \begin{bmatrix} -y^f(t) \\ \vdots \\ -\delta^{n-1}y^f(t) \\ u^f(t) \\ \vdots \\ \delta^m u^f(t) \end{bmatrix} \qquad \theta_0 = \begin{bmatrix} a_{\delta,0} \\ \vdots \\ a_{\delta,n-1} \\ b_{\delta,0} \\ \vdots \\ b_{\delta,m} \end{bmatrix} \tag{2.201}$$

All signals in $\varphi(t)$ are known at time t. The problem of estimating δ-domain transfer functions has now been transformed to a problem of estimating a linear regression model which is a well-known discipline, (Ljung 1987). Note, that the noise sequence becomes colored because of the filtering operation. This illustrates, that an ARX-model formulated in the δ-domain is not equivalent to a q-domain ARX-model. If the noise level is large and normal linear regression estimation is used then the estimates become biased.

It is worth mentioning, that using the perhaps obvious filter $1/F(\delta) = 1/\delta^n$ is not possible, since this filter is not asymptotically stable. More details on the choice of prefilter appear in the following.

2.7.2 Least Squares models

As pointed out in the previous section the δ-domain ARX model is not a least squares model. In the q-domain a least squares model is given by

$$A_q(q)y(t) = B_q(q)u(t) + q^n e(t) \tag{2.202}$$

Using (2.78) yields

$$\frac{1}{T^n}A_q(1+T\delta)y(t) = \frac{1}{T^n}B_q(1+T\delta)u(t) + \frac{1}{T^n}(1+T\delta)^n e(t) \tag{2.203}$$

or

$$A_\delta(\delta)y(t) = B_\delta(\delta)u(t) + (\delta + \frac{1}{T})^n e(t) \tag{2.204}$$

To obtain a white noise error in the regression form we can use $F(\delta) = (\delta + \frac{1}{T})^n$ which corresponds to the classical prefilter $1/F(q) = 1/q^n$ in the q-domain.

2.7.3 δ-domain ARMAX models

Consider now the δ-domain ARMAX model

$$A_\delta(\delta)y(t) = B_\delta(\delta)u(t) + C_\delta(\delta)e(t) \tag{2.205}$$

where $\{e(t)\}$ is a white noise sequence and

$$C_\delta(\delta) = \delta^n + c_{\delta,n-1}\delta^{n-1} + \cdots + c_{\delta,0} \tag{2.206}$$

Using the prefilter $1/F(\delta) = 1/C_\delta(\delta)$ will ensure, that the noise sequence is white, if estimation of $B_\delta(\delta)$ and $A_\delta(\delta)$ in an ARX structure is used, see

(2.200). However, this requires a priori knowledge of the spectral distribution of the noise which is generally not available.

Alternatively the noise polynomial $C_\delta(\delta)$ can be incorporated in the estimation problem by describing the model on a non-linear regression form using a general prefilter $1/F(\delta)$. Consider the rewriting

$$y(t) = \frac{C_\delta(\delta) - A_\delta(\delta)}{C_\delta(\delta)} y(t) + \frac{B_\delta(\delta)}{C_\delta(\delta)} u(t) + e(t) \tag{2.207}$$

This model is nonlinear in the parameters and consequently leads to a nonlinear regression problem where the regressor signals are obtained by filtering with the estimate of $1/C_\delta(\delta)$, see (Middleton and Goodwin 1990, Goodwin and Sin 1984). This may lead to local minima and in addition measures need to be taken to keep the estimate of $C_\delta(\delta)$ stable. Another way to reduce the bias of the transfer function estimate $B_\delta(\delta)/A_\delta(\delta)$ due to the colored noise sequence is looked into in the next section.

2.7.4 Regressor filtering

The design of prefilters in estimation where the intended use of the model is control design has become an important field of research in recent years. With low-complexity models the frequency distribution of the discrepancy between the estimated model and the true system can be affected by the prefilter. This so-called bias distribution is important for subsequent control design and hence, the prefilter should be chosen in a control-relevant way to improve the quality of the controller. These aspects of prefilter design are discussed in Chapter 5.

The use of regressor filters is a necessary and natural part of estimation in δ-operator parametrizations. This may not appear to be the case in q-domain parametrizations where the usual procedure is to wait for the signals in the regressor vector to become available. This actually corresponds to using the prefilter $F(q) = q^n$. As illustrated in (Middleton and Goodwin 1990) this is usually not a good choice.

An alternative to the nonlinear regression problem in estimation of ARMAX models with unknown noise polynomial $C_\delta(\delta)$ is investigated in (Middleton and Goodwin 1990). Here an overparametrization of the estimated deterministic part of the system is introduced to reduce the effect of colored noise. Only the deterministic transfer function B_δ/A_δ is estimated.

Theorem 7 *(Middleton and Goodwin 1990)*

Consider the filtered δ-domain ARMAX model

$$A_\delta(\delta)\frac{y(t)}{F(\delta)} = B_\delta(\delta)\frac{u(t)}{F(\delta)} + C_\delta(\delta)\frac{e(t)}{F(\delta)} \tag{2.208}$$

Provided we know an f, $0 < f \leq 1/T$, such that the zeros ξ_i of $C_\delta(\delta)$ satisfy

$$|\xi_i + f| < f \qquad \textit{for } i = 1, \ldots, n \tag{2.209}$$

then given any $\epsilon > 0$ there exists a system-independent polynomial $F'(\delta)$ such that the model can be expressed as

$$A'_\delta(\delta)\frac{y(t)}{F'(\delta)} = B'_\delta(\delta)\frac{u(t)}{F'(\delta)} + e(t) + e'(t) \tag{2.210}$$

where $B'_\delta(\delta)/A'_\delta(\delta)$ is not necessarily minimal and the variance of the filtered white noise sequence $e'(t)$ is less than or equal to ϵ. □

The theorem implies, that the filtered model can be made arbitrarily close to a model with white noise so that ordinary linear least-squares methods can be used on the filtered model.

As shown in (Middleton and Goodwin 1990) with no repeated zeros in $C_\delta(\delta)$ we can choose

$$F'(\delta) = F(\delta)(\delta + f)^p \tag{2.211}$$

where $F(\delta)$ can be any stable polynomial of degree n. The variance of e' can be made arbitrarily small by selecting p high enough. This choice of filter leads to an overparametrization in the estimation problem as

$$A'_\delta(\delta) = A_\delta(\delta)E'(\delta) \tag{2.212}$$
$$B'_\delta(\delta) = B_\delta(\delta)E'(\delta) \tag{2.213}$$

where $E'(\delta)$ is a monic polynomial of degree p. The polynomial $F(\delta)$ should be chosen as close as possible to $C_\delta(\delta)$ as the overparametrization needed then will be small, see (Middleton and Goodwin 1990).

The stability region of the δ-operator is described by

$$|\delta + \frac{1}{T}| < \frac{1}{T} \tag{2.214}$$

Hence, with $f = 1/T$ in the δ-plane the condition of the theorem is always satisfied. This choice of f corresponds to selecting $f = 0$ in the q-plane giving rise to the filter $F(q) = q^n$ which in turn corresponds to estimation in an unfiltered high-order ARX-structure. It is well-known that high-order ARX models are capable of approximating any linear system arbitrarily well (Ljung 1987). However, as illustrated in (Middleton and Goodwin 1990) $F(q) = q^n$ is usually not a good choice as the amount of overparametrization necessary is large if $F(\delta)$ is not close to $C_\delta(\delta)$.

2.8 Advantages by using δ-operator parametrizations

We will now briefly summarize some of the advantages, that can be obtained by using discrete-time models parametrized by the δ-operator rather than the usual shift-operator.

We have already discussed to following aspects:

- *Limiting properties at fast sampling rates:* The δ-domain description converges to its continuous-time counterpart as $T \to 0$; this holds for both state-space and transfer function representations.

- *Unification of continuous-time and discrete-time theory:* As we have seen in the optimal filtering problems the use of δ-operator parametrizations allows a unified theory for discrete-time and continuous-time results to be developed where the continuous-time results are obtained simply by letting $T \to 0$. Such a unified theory has been presented in (Middleton and Goodwin 1990).

- *A priori physical knowledge:* The transfer function coefficients of δ-operator models converge to the continuous-time coefficients as $T \to 0$. If the continuous-time model is derived from physical considerations (i.e. using first principles modelling) then this will often lead to some information on the continuous-time transfer function coefficients. With q-domain models this information is scattered in a complex way on the discrete-time coefficients which makes it very difficult to utilize such a priori information in for instance parameter estimation. Opposed to this the limiting behaviour of δ-operator models makes it much easier to incorporate a priori physical knowledge.

- *Non-minimumphase zeros:* In many control designs, for instance minimum variance control, one needs to pay special attention to non-minimumphase zeros. As we have seen non-minimumphase zeros are very likely to appear in sampled descriptions at fast sampling rates even when the continuous-time system is minimumphase. However, the structure of the numerator polynomial in δ-operator models makes it possible to avoid the troublesome sampling zeros by making certain approximations.

In addition to this, some observations concerning parameter estimation in δ-operator models are made in (Gevers and Li 1993). They find some interesting properties of the information matrix in least-squares estimation problems. For narrow-band input signals the information matrix is better conditioned compared to q-domain formulations and has a special structure with approximately all zeros in odd-off diagonal elements (for auto-regressive and moving average models). Both these properties are argued to be important for the convergence speed of adaptive estimation algorithms and to the numerical robustness. Some additional investigations concerning these properties of δ-domain parametrizations in estimation have been presented in (Rostgaard 1995).

Another aspect is the numerical properties of δ-operator parametrizations. This has been thoroughly investigated in (Gevers and Li 1993) where the following important points are mentioned.

- Superior finite word length coefficient representation
- Superior finite word length rounding error performance
- Superior numerical properties in calculation with discrete-time model, e.g. control design

We now briefly state some of the results obtained for δ-operator parametrizations in (Li and Gevers 1993, Gevers and Li 1993) for the two first points of the above list. Other results on finite word length implementations are available in (Middleton and Goodwin 1990) where also numerical issues in solving the optimal filtering problems are examined, i.e. finding the solution to the discrete-time Riccati difference equations.

2.8.1 Finite word length sensitivity of transfer functions

Consider the generalized state space realization

$$\rho x(t) = \boldsymbol{A}_\rho x(t) + \boldsymbol{B}_\rho u(t) \tag{2.215}$$

$$y(t) = \boldsymbol{C}_\rho x(t) \tag{2.216}$$

where the operator ρ is either q or δ. The corresponding transfer function is given by

$$G_\rho(\rho) = \boldsymbol{C}_\rho(\rho\boldsymbol{I} - \boldsymbol{A}_\rho)^{-1}\boldsymbol{B}_\rho \tag{2.217}$$

Define the real-valued positive sensitivity measure

$$M_{\rho,L_{i2}} = \left\| \frac{\partial G_\rho(\rho)}{\partial \boldsymbol{A}_\rho} \right\|_i^2 + \left\| \frac{\partial G_\rho(\rho)}{\partial \boldsymbol{B}_\rho} \right\|_2^2 + \left\| \frac{\partial G_\rho(\rho)}{\partial \boldsymbol{C}_\rho} \right\|_2^2 \tag{2.218}$$

with

$$\frac{\partial G_\rho(\rho)}{\partial \boldsymbol{A}_\rho} = (\rho\boldsymbol{I} - \boldsymbol{A}_\rho^\top)^{-1}\boldsymbol{C}_\rho^\top\boldsymbol{B}_\rho^\top(\rho\boldsymbol{I} - \boldsymbol{A}_\rho^\top)^{-1} \tag{2.219}$$

$$\frac{\partial G_\rho(\rho)}{\partial \boldsymbol{B}_\rho} = (\rho\boldsymbol{I} - \boldsymbol{A}_\rho^\top)^{-1}\boldsymbol{C}_\rho^\top \tag{2.220}$$

$$\frac{\partial G_\rho(\rho)}{\partial \boldsymbol{C}_\rho} = (\rho\boldsymbol{I} - \boldsymbol{A}_\rho)^{-1}\boldsymbol{B}_\rho \tag{2.221}$$

and where $i = 1, 2$ giving rise to either a mixed L_1/L_2 measure or a pure L_2 measure. The L_i norm of a complex-valued matrix function is defined as

$$\|f(\rho)\|_i = \left(\frac{1}{2\pi}\int_0^{2\pi} \|f(e^{j\omega})\|_F^i\right)^{1/i} \tag{2.222}$$

where $\|\cdot\|_F$ is the Frobenius norm. These sensitivity measures quantify the deviation of the transfer function with respect to changes in the coefficient of the state-space system matrices. The mixed L_1/L_2 measure give rise to sensitivity optimization problems with easily computable solutions while the pure L_2 measure perhaps is more natural, (Gevers and Li 1993).

Consider now two realizations $(\boldsymbol{A}_q, \boldsymbol{B}_q, \boldsymbol{C}_q)$ and $(\boldsymbol{A}_\delta, \boldsymbol{B}_\delta, \boldsymbol{C}_\delta)$ of the same transfer function where $\boldsymbol{A}_q = \boldsymbol{I} + T\boldsymbol{A}_\delta$, $\boldsymbol{B}_q = T\boldsymbol{B}_\delta$ and $\boldsymbol{C}_q = \boldsymbol{C}_\delta$. Then it follows that

$$\frac{\partial G_\delta(\delta)}{\partial \boldsymbol{A}_\delta} = T\frac{\partial G_q(q)}{\partial \boldsymbol{A}_q} \tag{2.223}$$

$$\frac{\partial G_\delta(\delta)}{\partial \boldsymbol{B}_\delta} = T\frac{\partial G_q(q)}{\partial \boldsymbol{B}_q} \tag{2.224}$$

$$\frac{\partial G_\delta(\delta)}{\partial \boldsymbol{C}_\delta} = \frac{\partial G_q(q)}{\partial \boldsymbol{C}_q} \tag{2.225}$$

which inserted in the definitions of the sensitivity measures gives

$$M_{\delta,L_{i2}} - M_{q,L_{i2}} = (T^2 - 1)\frac{\partial G_q(q)}{\partial \boldsymbol{A}_q} + (T^2 - 1)\frac{\partial G_q(q)}{\partial \boldsymbol{B}_q} \tag{2.226}$$

Clearly, the right-hand side of this equation is negative if $T < 1$. We now have the following theorem.

Theorem 8 *(Li and Gevers 1993)*

With the assumptions given above the following holds

$$M_{\delta,L_{12}} < M_{q,L_{12}} \quad \textit{and} \quad M_{\delta,L_{22}} < M_{q,L_{22}} \tag{2.227}$$

if and only if $T < 1$. □

That is, with $T < 1$ the δ-operator parametrizations always offer better performance with respect to the sensitivity of the transfer function to numerical errors in the coefficients of the state-space system matrices. In (Li and Gevers 1993) the variable T is considered to be a scaling parameter and not necessarily related to the sampling period. It is argued, that at fast sampling it is almost always possible to choose this scaling paremeter less than one.

In (Middleton and Goodwin 1990) the sensitivity of the pole and zero locations of transfer functions with respect to errors in the transfer function coefficients is analyzed. They reach similar conclusions, namely that δ-domain transfer functions are generally less sensitive to coefficient errors.

Truncated q-domain and δ-domain descriptions

Example 1 *As we have seen in Section 2.6 when the sampling period T tends to zero the q-domain system description becomes singular. All poles* *converge to $z = 1$ and except for the sampling zeros all zeros also converge to $z = 1$. This means, that the lower T the more numerical precision is required in order not to loose information about the system dynamics.*

On the other hand a δ-domain system description converges to the corresponding continuous-time description when $T \to 0$. Hence the numerical requirements do not increase when T tends to zero.

In Section 2.8.1 some theoretical results supported these observations also for the sensitivity of transfer functions with respect to numerical errors in the coefficients of their state-space representations. A simple illustration of the advantage of δ-operator parametrization in transfer function representation can be obtained as follows. Consider the stable minimum phase continuous-time system

$$\frac{B_s(s)}{A_s(s)} = \frac{s+2}{(s+1)(s^2+s+1)} = \frac{s+2}{s^3+2s^2+3s+2} \tag{2.228}$$

When sampled with a sampling period of $T = 0.1$ and a zero order hold on the input the discrete descriptions become

$$\frac{B_q(q)}{A_q(q)} = 10^{-3}\frac{4.99\,q^2 + 0.903\,q - 4.08}{q^3 - 2.79\,q^2 + 2.61\,q - 0.819} \tag{2.229}$$

$$\frac{B_\delta(\delta)}{A_\delta(\delta)} = \frac{0.0499\,\delta^2 + 1.09\,\delta + 1.81}{\delta^3 + 2.09\,\delta^2 + 2.99\,\delta + 1.81} \tag{2.230}$$

where all coefficients have been chopped to three significant digits. Note how $A_q(q)$ resembles the polynomial $(q-1)^3 = q^3 - 3q^2 + 3q - 1$ while $A_\delta(\delta)$ resembles $A_s(s)$. The poles of the original continuous-time system are given by

$$s = -0.5000 \pm 1.3229\,i \quad \text{and} \quad s = -1 \tag{2.231}$$

These should be compared to the poles of the truncated discrete-time system converted to the Laplace-domain by $s = \frac{1}{T}\ln(z)$ which become

$$s_q = -0.7730 \pm 1.3623\,i \quad \text{and} \quad s_q = -0.4508 \tag{2.232}$$

$$s_\delta = -0.4992 \pm 1.3259\,i \quad \text{and} \quad s_\delta = -0.9971 \tag{2.233}$$

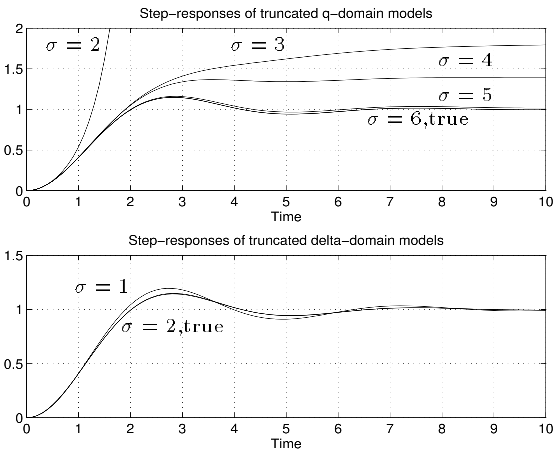

Figure 2.6: *Step-responses of true system and truncated systems with varying number of significant digits, q-domain case (top), δ-domain case (bottom).*

in the q-domain and δ-domain case respectively. It is clear, that the q-domain description suffers most from the truncation of the system polynomials. This is also clearly illustrated in Figure 2.6 where step-responses of truncated models in the two domains are compared to the true step-response with varying number of significant digits, σ. The δ-domain model captures the true response quite accurately with $\sigma = 2$ while the q-domain model requires $\sigma = 6$ to give approximately the same precision.

Note that this example is by no means outstanding. Truncated δ-domain models will always outperform the corresponding truncated q-domain models if T is chosen small enough. The example points out, that for instance optimal prediction based on δ-operator input-output relations as considered in Chapter 3 is preferable to shift-operator parametrizations with respect to finite word length problems. This is, as shown, simply due to the fact, that q-domain models loose much more information about the system dynamics when truncated than δ-domain models. It is also important to realize that if *the system model is obtained by an estimation algorithm then this will give similar uncertainty on the transfer function parameters. Therefore, identification in the δ-domain is likely to significantly improve the precision of subsequent model-based calculations, for instance prediction or control design.* □

2.8.2 Roundoff noise gain

Another aspect in implementation of state-space realizations of a given model is the effect of roundoff noise. This has also been investigated in (Gevers and Li 1993). Consider again the generalized state space realization

$$\rho x(t) = \boldsymbol{A}_\rho x(t) + \boldsymbol{B}_\rho u(t) \tag{2.234}$$

$$y(t) = \boldsymbol{C}_\rho x(t) \tag{2.235}$$

where the operator ρ is either q or δ. Assume, that the coefficients of the state-space matrices are represented exactly by B_c bits with fixed-point arithmetic and that input signal by B_s bits. The actual state and output are then represented by $B_c + B_s$ bits. We have the finite word length implementation

$$\rho x^*(t) = \boldsymbol{A}_\rho Q[x^*(t)] + \boldsymbol{B}_\rho u(t) \tag{2.236}$$

$$y^*(t) = \boldsymbol{C}_\rho Q[x^*(t)] \tag{2.237}$$

where $Q[\cdot]$ is the quantizer which reduces the representation of $x^*(t)$ to B_s bits before multiplication. The roundoff noise

$$e^*(t) = x^*(t) - Q[x^*(t)] \tag{2.238}$$

is modelled as zero-mean white noise with variance $u^2\boldsymbol{I}$ where $u^2 = 2^{-2B_s}/12$. Define the roundoff noise gain as

$$K_\rho = \frac{1}{u^2} \lim_{t\to\infty} \mathcal{E}\left\{(y(t) - y^*(t))^2\right\} \tag{2.239}$$

The roundoff noise gain is a measure of the ratio between the variance of the output error and the variance of the roundoff noise.

Without going into details we now give one result from (Gevers and Li 1993). Denote the minimum achievable roundoff noise gain of q and δ operator models in a given set of realizations as $K_q^{\min}$ and $K_\delta^{\min}$, respectively.

Theorem 9 *(Gevers and Li 1993)*

For any stable minimal discrete-time SISO linear time-invariant system, if the q-domain poles λ_i of the system satisfy

$$\sum_{i=1}^{n} \lambda_i \geq n - \frac{1}{2} \tag{2.240}$$

then

$$K_\delta^{min} < K_q^{min} \tag{2.241}$$

□

As we have seen in Section 2.6.3 the poles of q-domain models cluster around the point $(1, 0)$ in the complex plane in case of fast sampling. Clearly, the condition of the above theorem is then always satisfied with sufficiently fast sampling. Hence, δ-operator implementations offer better performance in terms of minimizing the roundoff noise gain. As illustrated by numerical examples in (Gevers and Li 1993) the advantage is often quite substantial.

2.9 Chapter conclusion

In this chapter we have discussed the mathematical modelling of linear stochastic systems. We have derived equivalent discrete-time models for the sampled versions of continuous-time stochastic system descriptions. Both q-domain and δ-domain parametrizations have been discussed. The δ-domain parametrizations offer several advantages to q-domain representations, especially at fast sampling rates. One important advantage is the fact, that the δ-domain framework for discrete-time modelling is closely related to continuous-time and in fact, the δ-domain models converge to their continuous-time counterparts as the sampling period tends to zero. These properties of the δ-domain modelling framework makes it possible to treat discrete-time and continuous-time theory in a unified manner. This has been illustrated for the optimal linear filtering problem. This is in contrast to the q-domain framework which displays misleading limiting properties as the sampling period tends to zero.

Transfer function representations for discrete-time models have been derived. Furthermore, explicit expressions for the connection between δ-domain and q-domain models have been found. The connection between continuous-time and δ-domain models also becomes apparent when considering the limiting properties of transfer function coefficients and pole/zero locations. The estimation of δ-domain transfer functions has been briefly discussed. A regressor filter is a necessary part of the estimation problem and some aspects on the choice of this filter are discussed.

In implementation of discrete-time models on digital computers with finite word length arithmetic we find that δ-domain parametrizations are superior to q-domain parametrizations. This is illustrated by a simple numerical example.

Chapter 3

Optimal and Emulator-Based Prediction

3.1 Introduction

We have seen in Chapter 2, that δ-domain models provide several advantages compared to q-domain representations in system modelling, especially at fast sampling rates. The advantages in face of finite word length implementations are passed on to calculations based on discrete-time models, e.g. control design. One particular type of controller, which has gained growing popularity in recent years, is long range predictive control as discussed in Chapter 4. One step in finding such controllers is prediction of the system output. In this chapter we derive expressions for long range output prediction using discrete-time δ-domain models.

A further profit from the use of δ-operator parametrizations is the close connection to continuous-time at fast sampling rates. As a step in the unification of discrete-time and continuous-time theory a new more general prediction method is developed in the present chapter. This method is based on the idea of time-derivative emulation in continuous-time as presented in (Demircioğlu and Gawthrop 1991) and has a well-defined limit as the sampling period tends to zero. The new method, the so-called δ-domain emulator-based predictor, uses discrete-time δ-domain models but has a capability to predict the output continuously. This opens up a possibility to consider continuous-time like criterion functions in subsequent long range predictive control design. This is discussed in Chapter 4. The δ-domain emulator-based predictor was first presented in (Lauritsen *et al.* 1994b, Lauritsen *et al.* 1995b).

When constructing a predictor one has to consider on which amount of information the predictor should be based. Two important cases arise.

- Complete information:
  $\mathcal{I}_{c,t} = \{y(t), y(t-T), \ldots, u(t-T), u(t-2T), \ldots\}$
- Incomplete information:
  $\mathcal{I}_{i,t} = \{y(t-T), y(t-2T), \ldots, u(t-T), u(t-2T), \ldots\}$

where $\mathcal{I}_{c,t}$ and $\mathcal{I}_{i,t}$ are sets of signal-values representing the available information and T is the sampling period. The only difference between the two cases is whether or not the latest measurement of the output $y(t)$ is known. In practical cases it may very well be the case, that this measurement can not be used, for instance if the response-time of a digital controller must be small.

In the following sections three different angles of attack to optimal prediction will be investigated using both the traditional shift-operator and the δ-operator. It will appear, that the relative degree of the noise model has influence on which information set should be used in the emulator-based approach to optimal prediction to avoid jeopardizing the close connection to the continuous-time case. The predictors will be presented on forms well-suited to a following inclusion in a long range predictive control scheme. In particular this means, that the effect of future input signals on the predictions is separated from the effect of past input.

As we are dealing with long range prediction with the aim of computing long range predictive controllers we also need to consider the future control input as known as this is our decision variable in the control optimization problem. That is, we will include the set

$$\mathcal{U}_{t+k} = \{u(t + (k - d)T), u(t + (k - d - 1)T), \ldots, u(t)\} \tag{3.1}$$

in our information set where d is the time-delay in samples of the system model. The k-step prediction of the output is of course affected only by input signals up to time $t + (k - d)T$. Let

$$\mathcal{I}_{i,t,k} = \mathcal{I}_{i,t} \cup \mathcal{U}_{t+k} \tag{3.2}$$

and correspondingly

$$\mathcal{I}_{c,t,k} = \mathcal{I}_{c,t} \cup \mathcal{U}_{t+k} \tag{3.3}$$

The chapter is organized as follows. Section 3.2 briefly reviews the derivation of optimal prediction based on q-domain ARMAX system descriptions. Both incomplete and complete information is considered. The prediction scheme is then translated to the δ-domain in Section 3.3. Similar work can be found in (Rostgaard *et al.* 1994a).

In Section 3.4 the new δ-domain emulator-based predictor is presented. The connection to the continuous-time predictor presented in (Demircioğlu and Gawthrop 1991) is shown and in addition the new predictor is shown to contain the optimal predictor as a special case. In Section 3.5 some simulation examples illustrating the properties of the δ-domain emulator-based predictor are presented.

3.2 Optimal prediction in the q-domain

Consider the q-domain ARMAX-model

$$A_q(q)y(t) = B_q(q)u(t) + C_q(q)e(t) \tag{3.4}$$

with the following polynomial orders

$$\deg(A_q) = n \tag{3.5}$$
$$\deg(B_q) = m < n \tag{3.6}$$
$$\deg(C_q) = n - \Delta \tag{3.7}$$

where $\Delta \in \{0, 1\}$, i.e. $C_q(q)/A_q(q)$ is either exactly proper or strictly proper. A discussion on the choice of Δ appears below. $C_q(q)$ is assumed to have all its roots inside the unit circle. This is justified by the spectral factorization lemma except for possible roots on the stability circle, see (Goodwin and Sin 1984). Without loss of generality $A_q(q)$ and $C_q(q)$ are assumed to be monic. The noise sequence $\{e(t)\}$ is assumed to be white noise. The time delay in samples is given by

$$d = n - m \tag{3.8}$$

A discrete-time model obtained by sampling with a zero-order hold will have $d \geq 1$, almost always with equality.

The coefficients of the A_q polynomial are defined by

$$A_q(q) = a_{q,n}q^n + a_{q,n-1}q^{n-1} + \cdots + a_{q,0} \tag{3.9}$$

The coefficients of all other polynomials occurring in the chapter are defined similarly. Note, that the monicness assumption for $A_q(q)$ implies $a_{q,n} = 1$.

3.2.1 Discussion on the degree of the noise polynomial

As we have seen in Chapter 2 a linear system with measurement noise acting directly on the output will always give rise to a corresponding ARMAX description with a noise polynomial of degree n, where n is the order of the system and where the driving noise input to the ARMAX model is the innovations sequence. This is regardless of the domain of the description, be it continuous-time, q-domain or δ-domain. Without measurement noise the degree of the noise polynomial will usually be $n-1$ for discrete-time systems corresponding to $\Delta = 1$, see (Rostgaard 1995). Hence, the most common choice in discrete-time modelling is $\Delta = 0$. Reducing the degree of the noise polynomial in the discrete-time model corresponds to introducing a delay in the noise transfer function or equivalently shifting the noise sequence in time. As the properties of the type of noise sequences considered here remain unchanged by time-shifts we can in fact transform the case $\Delta = 1$ to $\Delta = 0$ by time-shifting the noise-sequence. This is commented further on in Section 3.2.5.

In continuous-time the situation is somewhat different. Consider the continuous-time ARMAX model

$$A_s(s)Y(s) = B_s(s)U(s) + C_s(s)E(s) \tag{3.10}$$

where s is the complex Laplace-operator and $Y(s)$, $U(s)$ and $E(s)$ are the transformed output, input and noise signals. The relative degree of the noise transfer function $C_s(s)/A_s(s)$ effects the roll-off at high frequencies. Often the noise polynomial is treated as a design parameter in the control design. Then the motivation behind the choice of the degree of the noise-polynomial may be different than that dictated by physical considerations. This is discussed in (Gawthrop 1987) where it is noted, that if the noise component $E(s)$ of the system contains high bandwidth white noise or impulsive behaviour it may not be desirable to differentiate it. This can be modelled by choosing $\deg(C_s) = n-1$. If we do not even wish to use the system output directly we must choose $\deg(C_s) = n$.

It appears, that the case $\Delta = 1$ is relevant in continuous-time models. As the emulator-based predictor presented later in the chapter tries to make a bridge between the two domains we will throughout the chapter allow for both choices by means of the variable Δ. The price to pay is a somewhat

Modelling case	Information set	Δ	Reconstructable noise signals
$\mathcal{M}_0$	Complete, $\mathcal{I}_{c,t,k}$	0	$e(t)$, $e(t-T)$,...
$\mathcal{M}_1$	Complete, $\mathcal{I}_{c,t,k}$	1	$e(t-T)$, $e(t-2T)$,...
$\mathcal{M}_2$	Incomplete, $\mathcal{I}_{i,t,k}$	0	$e(t-T)$, $e(t-2T)$,...
$\mathcal{M}_3$	Incomplete, $\mathcal{I}_{i,t,k}$	1	$e(t-2T)$, $e(t-3T)$,...

Table 3.1: Reconstructable noise signals for varying information set and system model.

more complex presentation. It should be mentioned, that the continuous-time emulator-based predictor presented in (Demircioğlu and Gawthrop 1991) on which we build the δ-domain version presented in Section 3.4 assumes $\deg(C_s) = n-1$. Hence, to get a continuous-time limit of the δ-domain emulator-based predictor which coincides with the continuous-time predictor from (Demircioğlu and Gawthrop 1991) it is convenient to treat both the case of $\Delta = 0$ and of $\Delta = 1$.

3.2.2 Information/model cases considered

The two information sets $\mathcal{I}_{i,t,k}$ and $\mathcal{I}_{c,t,k}$ and the two possible degrees of the noise polynomial make a total of four possible cases for which optimal predictor expressions can be developed. An important characteristic of each case is how much information about old noise data can be extracted from the available information and the system model. This is illustrated for all cases in Table 3.1.

The table also defines symbols for the modelling cases. Complete information with $\Delta = 0$ is denoted by $\mathcal{M}_0$. This is the case usually considered when designing discrete-time optimal predictors. As discussed previously $\Delta = 0$ is the most natural choice of model in discrete-time and as will be shown in Section 3.2.4, the variance of the prediction error is reduced when complete information is used. However, it will later in Section 3.4.3 turn out, that for the new emulator-based predictor the close connection to continuous-time can only be preserved in two cases, namely complete information with $\Delta = 1$ ($\mathcal{M}_1$) and incomplete information with $\Delta = 0$ ($\mathcal{M}_2$). For this reason emphasis is put on these two cases in the rest of the chapter. We will handle both cases simultaneously. This is possible as they give rise to very similar expressions for output prediction due to the fact, that the same set of old noise signals can be reconstructed from input-output data as illustrated in Table 3.1. The simultaneous treatment is eased by introducing the model dependent information set

$$\boxed{\mathcal{I}_{\Delta,t,k} \triangleq \begin{cases} \mathcal{I}_{c,t,k} & \text{if } \Delta = 1 \\ \mathcal{I}_{i,t,k} & \text{if } \Delta = 0 \end{cases} \qquad (3.11)}$$

We pay less attention to modelling cases $\mathcal{M}_0$ and $\mathcal{M}_3$. We briefly show the connection between the cases $\mathcal{M}_0$ and $\mathcal{M}_2$, i.e. complete and incomplete information with $\deg(C_q) = n$. Modelling case $\mathcal{M}_3$, incomplete information with $\deg(C_q) = n - 1$, can be regarded as a special case of $\mathcal{M}_2$, incomplete information with $\deg(C_q) = n$ as shown in Section 3.2.5. In general one would prefer $\mathcal{M}_1$ to $\mathcal{M}_3$ as the variance of the prediction error is reduced with complete information.

3.2.3 Modelling cases $\mathcal{M}_1$ and $\mathcal{M}_2$.

As these two cases are the only ones relevant for the emulator-based predictor which is introduced later it is convenient to treat them simultaneously. Therefore, in this section we work with the information set $\mathcal{I}_{\Delta,t,k}$ which depends on the chosen model structure.

To derive the optimal predictor consider[1] the system output at time $t+kT$

$$y(t+kT) = \frac{B_q}{A_q} u(t+kT) + \frac{C_q}{A_q} e(t+kT) \tag{3.12}$$

By means of the Diophantine-equation

$$q^k \frac{C_q(q)}{A_q(q)} = E_q^{[k]}(q) + \frac{F_q^{[k]}(q)}{A_q(q)} \tag{3.13}$$

with order restrictions

$$\deg(E_q^{[k]}) \quad = \quad \max\{k-\Delta, 0\} \tag{3.14}$$

$$\deg(F_q^{[k]}) \quad = \quad n-1 \tag{3.15}$$

we can split up the noise term in a term consisting of future noise values and another one consisting of past noise values, i.e.

$$\frac{C_q}{A_q} e(t+kT) = E_q^{[k]} e(t) + \frac{F_q^{[k]}}{A_q} e(t) \tag{3.16}$$

The past noise values are not directly available from $\mathcal{I}_{\Delta,t,k}$, however if we assume a correct system model then they can be reconstructed from measurable data, see Table 3.1

$$\frac{F_q^{[k]}}{A_q} e(t) = \frac{F_q^{[k]}}{A_q C_q} \left(A_q y(t) - B_q u(t) \right) \tag{3.17}$$

Insertion into (3.12) yields

$$y(t+kT) = \frac{F_q^{[k]}}{C_q} y(t) + \left(1 - q^{-k} \frac{F_q^{[k]}}{C_q} \right) \frac{B_q}{A_q} u(t+kT) + E_q^{[k]} e(t) \tag{3.18}$$

Using (3.13) now gives

$$y(t+kT) = \frac{F_q^{[k]}}{C_q} y(t) + \frac{B_q E_q^{[k]}}{C_q} u(t) + E_q^{[k]} e(t) \tag{3.19}$$

[1]The argument of the polynomials will be left out in the rest of the chapter where convenient.

By means of the Diophantine equation

$$\frac{B_q(q)E_q^{[k]}(q)}{C_q(q)} = G_q^{[k]}(q) + \frac{H_q^{[k]}(q)}{C_q(q)} \tag{3.20}$$

with the order restrictions

$$\deg(G_q^{[k]}) = \max\{k-d, 0\} \tag{3.21}$$

$$\deg(H_q^{[k]}) = \min\{n-1-\Delta, m+\deg(E_q^{[k]})\} \tag{3.22}$$

we can separate the effect of the input signal in a term giving the effect of future inputs and a term consisting of past input data. Then the model can be stated on the form

$$y(t+kT) = \underbrace{G_q^{[k]}u(t)}_{\text{future}} + \underbrace{\frac{H_q^{[k]}}{C_q}u(t) + \frac{F_q^{[k]}}{C_q}y(t)}_{\text{known}} + \underbrace{E_q^{[k]}e(t)}_{\text{future}} \tag{3.23}$$

Notice, that the noise term $E_q^{[k]}e(t)$ consists exactly by noise signals independent from the information contained in $\mathcal{I}_{\Delta,t,k}$. Hence, the minimum-variance predictor is obtained by taking conditional expectation based on the information given in $\mathcal{I}_{\Delta,t,k}$. Define

$$\hat{y}(t+kT|\mathcal{I}_{\Delta,t,k}) \triangleq \mathcal{E}\{y(t+kT)|\mathcal{I}_{\Delta,t,k}\} \tag{3.24}$$

Then

$$\hat{y}(t+kT|\mathcal{I}_{\Delta,t,k}) = G_q^{[k]}u(t) + \frac{H_q^{[k]}}{C_q}u(t) + \frac{F_q^{[k]}}{C_q}y(t) \tag{3.25}$$

with the stationary prediction error $E_q^{[k]}e(t)$.

The two coupled Diophantine equations (3.13) and (3.20) are not general Diophantine equation but can both be solved simply by polynomial long division. This enables a recursive solution. For $j = 0, 1, \ldots, k$ the two coupled Diophantine equations can be solved recursively by the algorithm listed in Table 3.2. The recursive algorithm is proved in Appendix A.

Table 3.2: *Algorithm to recursively solve Diophantine-equations for q-domain optimal predictor for modelling cases $\mathcal{M}_1$ and $\mathcal{M}_2$.*

- *Initialization*

  1. $E_q^{[0]} = \begin{cases} 1 & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

  2. $F_q^{[0]} = \begin{cases} C_q - A_q & \text{if } \Delta = 0 \\ C_q & \text{if } \Delta = 1 \end{cases}$

  3. $G_q^{[0]} = \begin{cases} 0 & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

  4. $H_q^{[0]} = \begin{cases} B_q & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

- *Recursion*

  1. $\alpha = f_{q,n-1}^{[j]}$

  2. $E_q^{[j+1]} = qE_q^{[j]} + \alpha$

  3. $F_q^{[j+1]} = qF_q^{[j]} - \alpha A_q$

  4. $\beta = \begin{cases} h_{q,n-1}^{[j]} & \text{if } \Delta = 0 \\ h_{q,n-2}^{[j]} + \alpha b_{q,n-1} & \text{if } \Delta = 1 \end{cases}$

  5. $G_q^{[j+1]} = qG_q^{[j]} + \beta$

  6. $H_q^{[j+1]} = qH_q^{[j]} - \beta C_q + \alpha B_q$

Notice the shifting structure of the updates of E_q^j and G_q^j in the recursive algorithm, i.e. the coefficients are all shifted to terms of one degree higher and a new coefficient to the degree zero term is introduced at each iteration.

3.2.4 Modelling case $\mathcal{M}_0$

The optimal predictor for modelling case $\mathcal{M}_0$, i.e complete information and $\deg(C_q) = n$ can easily be derived from the above. Let

$$E_q^{[k]} = e_{q,0} + q\overline{E}_q^{[k]} \tag{3.26}$$

Then

$$\begin{aligned} y(t+kT) &= G_q^{[k]}u(t) + \frac{H_q^{[k]}}{C_q}u(t) \\ &\quad + \frac{F_q^{[k]}}{C_q}y(t) + e_{q,0}e(t) + \overline{E}_q^{[k]}e(t+T) \end{aligned} \tag{3.27}$$

Using the fact, that

$$e(t) = \frac{A_q}{C_q}y(t) - \frac{B_q}{C_q}u(t) \tag{3.28}$$

we get

$$\begin{aligned} y(t+kT) &= G_q^{[k]}u(t) + \frac{H_q^{[k]} - e_{q,0}B_q}{C_q}u(t) \\ &\quad + \frac{F_q^{[k]} + e_{q,0}A_q}{C_q}y(t) + \overline{E}_q^{[k]}e(t+T) \end{aligned} \tag{3.29}$$

so, that the optimal predictor based on complete information becomes

$$\begin{aligned} \hat{y}(t+kT|\mathcal{I}_{c,t,k}) &= G_q^{[k]}u(t) + \frac{H_q^{[k]} - e_{q,0}B_q}{C_q}u(t) + \frac{F_q^{[k]} + e_{q,0}A_q}{C_q}y(t) \\ &= G_q^{[k]}u(t) + \frac{\overline{H}_q^{[k]}}{C_q}u(t) + q\frac{\overline{F}_q^{[k]}}{C_q}y(t) \end{aligned} \tag{3.30}$$

with stationary prediction error $\overline{E}_q^{[k]} e(t+T)$, where

$$\begin{aligned} \overline{H}_q^{[k]} &= H_q^{[k]} - e_{q,0} B_q && (3.31) \\ q\overline{F}_q^{[k]} &= F_q^{[k]} + e_{q,0} A_q && (3.32) \end{aligned}$$

Alternatively, the two coupled Diophantine equations

$$\begin{aligned} q^{k-1} \frac{C_q(q)}{A_q(q)} &= \overline{E}_q^{[k]}(q) + \frac{\overline{F}_q^{[k]}(q)}{A_q(q)} && (3.33) \\ \frac{qB_q(q)\overline{E}_q^{[k]}(q)}{C_q(q)} &= G_q^{[k]}(q) + \frac{\overline{H}_q^{[k]}(q)}{C_q(q)} && (3.34) \end{aligned}$$

with order restrictions

$$\begin{aligned} \deg(\overline{E}_q^{[k]}) &= \max\{k-1, 0\} && (3.35) \\ \deg(\overline{F}_q^{[k]}) &= n-1 && (3.36) \\ \deg(\overline{G}_q^{[k]}) &= \max\{k-d, 0\} && (3.37) \\ \deg(\overline{H}_q^{[k]}) &= \min\{n-1, m + \deg(\overline{E}_q^{[k]})\} && (3.38) \end{aligned}$$

could be solved, again yielding the optimal predictor based on complete information given in (3.30).

The variance of the prediction error is reduced when complete information is used since

$$V\{E_q^{[k]} e(t)\} = V\{\overline{E}_q^{[k]} e(t)\} + e_{q,0}^2 V\{e(t)\} \qquad (3.39)$$

This is why modelling case $\mathcal{M}_0$ is generally preferred to modelling case $\mathcal{M}_2$. However, we are also concerned with the connection to continuous-time and in this respect modelling case $\mathcal{M}_2$ more suitable in our application.

3.2.5 Relating the cases of $\Delta = 0$ and $\Delta = 1$

Consider the model

$$A_q(q)y(t) = B_q(q)u(t) + C_q(q)e(t) \tag{3.40}$$

where $\deg(C_q) = n - 1$. Introduce the new noise polynomial

$$C'_q(q) = qC_q(q) \tag{3.41}$$

so that $\deg(C'_q) = n$. Then

$$A_q(q)y(t) = B_q(q)u(t) + C'_q(q)q^{-1}e(t) \tag{3.42}$$

or

$$A_q(q)y(t) = B_q(q)u(t) + C'_q(q)v(t) \tag{3.43}$$

where the new noise signal $v(t) = e(t - T)$. This model is exactly equivalent to the usual ARMAX model with $\deg(C_q) = n$. Hence, for both complete and incomplete information the optimal predictor expressions for $\Delta = 1$ can be considered special cases of $\Delta = 0$. The circumstances are different in continuous-time, as the relative degree of the noise transfer function then does not reflect a time-shift of the noise sequence but effects the roll-off at high frequencies of the disturbance acting on the output.

3.2.6 Summarizing q-domain optimal prediction

An overview of the different cases of optimal prediction in the q-domain is given below.

Modelling case $\mathcal{M}_0$

Diophantine equations:

$$q^{k-1}\frac{C_q(q)}{A_q(q)} = \overline{E}(q) + \frac{\overline{F}(q)}{A_q(q)} \tag{3.44}$$

$$\frac{qB_q(q)\overline{E}(q)}{C_q(q)} = \overline{G}(q) + \frac{\overline{H}(q)}{C_q(q)} \tag{3.45}$$

Order restrictions:

$$\begin{aligned}
\deg(\overline{E}) &= \max\{k-1,0\} & (3.46)\\
\deg(\overline{F}) &= n-1 & (3.47)\\
\deg(\overline{G}) &= \max\{k-d,0\} & (3.48)\\
\deg(\overline{H}) &= \min\{n-1, m+\deg(\overline{E})\} & (3.49)
\end{aligned}$$

Optimal k-step prediction:

$$\hat{y}(t+kT|\mathcal{I}_{c,t,k}) = \overline{G}(q)u(t) + \frac{\overline{H}(q)}{C_q(q)}u(t) + \frac{q\overline{F}(q)}{C_q(q)}y(t) \tag{3.50}$$

with stationary prediction error $\overline{E}(q)e(t+T)$.

Modelling cases $\mathcal{M}_1$ and $\mathcal{M}_2$

Diophantine equations:

$$\begin{aligned}
q^k\frac{C_q(q)}{A_q(q)} &= E(q) + \frac{F(q)}{A_q(q)} & (3.51)\\
\frac{B_q(q)E(q)}{C_q(q)} &= G(q) + \frac{H(q)}{C_q(q)} & (3.52)
\end{aligned}$$

Order restrictions:

$$\begin{aligned}
\deg(E) &= \max\{k-\Delta,0\} & (3.53)\\
\deg(F) &= n-1 & (3.54)\\
\deg(G) &= \max\{k-d,0\} & (3.55)\\
\deg(H) &= \min\{n-\Delta-1, m+\deg(E)\} & (3.56)
\end{aligned}$$

Optimal k-step prediction:

$$\hat{y}(t+kT|\mathcal{I}_{\Delta,t,k}) = G(q)u(t) + \frac{H(q)}{C_q(q)}u(t) + \frac{F(q)}{C_q(q)}y(t) \tag{3.57}$$

with stationary prediction error $E(q)e(t)$.

Modelling case $\mathcal{M}_3$

Diophantine equations:

$$q^{k+1}\frac{C_q(q)}{A_q(q)} = \underline{E}(q) + \frac{\underline{F}(q)}{A_q(q)} \tag{3.58}$$

$$\frac{B_q(q)\underline{E}(q)}{qC_q(q)} = \underline{G}(q) + \frac{\underline{H}(q)}{qC_q(q)} \tag{3.59}$$

Order restrictions:

$$\deg(\underline{E}) = k \tag{3.60}$$

$$\deg(\underline{F}) = n-1 \tag{3.61}$$

$$\deg(\underline{G}) = \max\{k-d, 0\} \tag{3.62}$$

$$\deg(\underline{H}) = \min\{n-1, m+k\} \tag{3.63}$$

Optimal k-step prediction:

$$\hat{y}(t+kT|\mathcal{I}_{i,t,k}) = \underline{G}(q)u(t) + \frac{\underline{H}(q)}{qC_q(q)}u(t) + \frac{\underline{F}(q)}{qC_q(q)}y(t) \tag{3.64}$$

with stationary prediction error $\underline{E}(q)e(t-T)$.

Notice, that

$$\overline{G}(q) = G(q) = \underline{G}(q) \tag{3.65}$$

This can be seen simply by noting, that with zero initial conditions the deterministic part of the output is given as $\overline{G}(q)u(t)$, $G(q)u(t)$ and $\underline{G}(q)u(t)$ in the three cases. That is, all three polynomials contain the impulse response parameters of the deterministic part of the plant model $B_q(q)/A_q(q)$.

3.3 Optimal prediction in the δ-domain

In this section we will work with the δ-domain ARMAX model

$$A_\delta(\delta)y(t) = B_\delta(\delta)u(t) + C_\delta(\delta)e(t) \tag{3.66}$$

where $A_\delta(\delta)$, $B_\delta(\delta)$ and $C_\delta(\delta)$ are polynomials in the δ-operator

$$\delta = \frac{q-1}{T} \tag{3.67}$$

Again A_δ and C_δ are assumed to be monic and C_δ is assumed to have all its roots inside the δ-domain stability circle.

As we have seen in Chapter 2 an exact equivalence to the q-domain ARMAX model can easily be established. For instance, in case $\Delta = 0$

$$A_\delta(\delta) = \frac{1}{T^n} A_q(1+\delta T) \tag{3.68}$$

$$B_\delta(\delta) = \frac{1}{T^n} B_q(1+\delta T) \tag{3.69}$$

$$C_\delta(\delta) = \frac{1}{T^n} C_q(1+\delta T) \tag{3.70}$$

The factor $1/T^n$ has been introduced to keep A_δ and C_δ monic.

The optimal predictor based on the information set $\mathcal{I}_{\Delta,t,k}$ for $\deg(C_\delta) = n - \Delta$ is derived in an analogous manner to the q-domain case simply by making the transformation $q = 1 + \delta T$. Hence, by means of the two coupled Diophantine-equations

$$(1+\delta T)^k \frac{C_\delta(\delta)}{A_\delta(\delta)} = E_\delta^{[k]}(\delta) + \frac{F_\delta^{[k]}(\delta)}{A_\delta(\delta)} \tag{3.71}$$

$$\frac{B_\delta(\delta) E_\delta^{[k]}(\delta)}{C_\delta(\delta)} = G_\delta^{[k]}(\delta) + \frac{H_\delta^{[k]}(\delta)}{C_\delta(\delta)} \tag{3.72}$$

with the order restrictions

$$\deg(E_\delta^{[k]}) = \max\{k-\Delta, 0\} \tag{3.73}$$

$$\deg(F_\delta^{[k]}) = n-1 \tag{3.74}$$

$$\deg(G_\delta^{[k]}) = \max\{k-d, 0\} \tag{3.75}$$

$$\deg(H_\delta^{[k]}) = \min\{n-1-\Delta, m + \deg(E_\delta^{[k]})\} \tag{3.76}$$

we can write the output at time $t + kT$ as

$$y(t + kT) = G_\delta^{[k]} u(t) + \frac{H_\delta^{[k]}}{C_\delta} u(t) + \frac{F_\delta^{[k]}}{C_\delta} y(t) + E_\delta^{[k]} e(t) \qquad (3.77)$$

Again the optimal predictor is obtained by taking conditional expectation based on $\mathcal{I}_{\Delta,t,k}$ which gives

$$\hat{y}(t + kT|\mathcal{I}_{\Delta,t,k}) = G_\delta^{[k]} u(t) + \frac{H_\delta^{[k]}}{C_\delta} u(t) + \frac{F_\delta^{[k]}}{C_\delta} y(t) \qquad (3.78)$$

For $j = 0, 1, \ldots, k$ the two coupled Diophantine equations can be solved recursively by the algorithm shown in Table 3.3 which is proved in Appendix A. Notice, that the updates of E_δ^j, F_δ^j, G_δ^j and H_δ^j become somewhat more complex compared to the corresponding updates from the q-domain case. In particular, the coefficients of E_δ^j and G_δ^j do not shift from one iteration to the next.

The case of complete information with $\Delta = 0$ ($\mathcal{M}_0$) and other modelling cases can also be handled by direct transformation from the q-domain case.

3.4 Emulator-based prediction

The two approaches to optimal prediction described in the previous sections are both based on the idea of predicting a number of future samples rather than predicting the output at an absolute future time instant. This means, that when $T \to 0$ then the algorithms become unrealizable. To obtain a predictor which behaves well when $T \to 0$, we will start out by looking at the continuous-time emulator-based predictor presented in (Demircioğlu and Gawthrop 1991). In continuous time we can approximate a function $y(t)$ in the neighborhood of t by the Taylor-series expansion

$$y(t + \tau) = y(t) + \sum_{j=1}^{N_y} \frac{\tau^j}{j!} \frac{d^j}{dt^j} y(t) + o(\tau^{N_y}) \qquad (3.79)$$

Table 3.3: *Algorithm to recursively solve Diophantine-equations for δ-domain optimal predictor in modelling cases $\mathcal{M}_1$ and $\mathcal{M}_2$.*

- *Initialization*

1. $E_\delta^{[0]} = \begin{cases} 1 & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

2. $F_\delta^{[0]} = \begin{cases} C_\delta - A_\delta & \text{if } \Delta = 0 \\ C_q & \text{if } \Delta = 1 \end{cases}$

3. $G_\delta^{[0]} = \begin{cases} 0 & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

4. $H_\delta^{[0]} = \begin{cases} B_\delta & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

- *Recursion*

1. $\alpha = T\, f_{\delta,n-1}^{[j]}$

2. $E_\delta^{[j+1]} = (1 + \delta T) E_\delta^{[j]} + \alpha$

3. $F_\delta^{[j+1]} = (1 + \delta T) F_\delta^{[j]} - \alpha A_\delta$

4. $\beta = \begin{cases} T\, h_{\delta,n-1}^{[j]} & \text{if } \Delta = 0 \\ T h_{\delta,n-2}^{[j]} + \alpha b_{\delta,n-1} & \text{if } \Delta = 1 \end{cases}$

5. $G_\delta^{[j+1]} = (1 + \delta T) G_\delta^{[j]} + \beta$

6. $H_\delta^{[j+1]} = (1 + \delta T) H_\delta^{[j]} - \beta C_\delta + \alpha B_\delta$

where $\tau \in \mathbb{R}$ is a real variable and $o(x)$ satisfies $\lim_{x\to 0} o(x)/x = 0$. In a matrix notation this can be restated as

$$y(t+\tau) = \boldsymbol{T}_{N_y}(\tau)\left[y(t)\ \frac{d}{dt}y(t)\ \ldots\ \frac{d^{N_y}}{dt^{N_y}}y(t)\right]^{\top} + o(\tau^{N_y}) \tag{3.80}$$

where

$$\boldsymbol{T}_{N_y}(\tau) = \left[1\ \tau\ \frac{\tau^2}{2!}\ \cdots\ \frac{\tau^{N_y}}{N_y!}\right] \tag{3.81}$$

This expression can be used to predict $y(t+\tau)$ for positive τ assuming knowledge of the derivatives of $y(t)$. With the close relationship between the δ-operator and time-derivation in mind we will formulate a δ-domain version of this continuous-time prediction principle.

3.4.1 Emulator-based prediction in the δ-domain

Knowing that the δ-operator converges to time-derivation as the sampling period tends to zero a δ-domain approximation of the above *may* be obtained simply by replacing d/dt with δ. However, knowledge of δ-derivatives up to order N_y is then used in the approximation of the output. Hence, better approximations of time-derivatives of lower order than N_y can be obtained if all N_y δ-derivatives are utilized. Such improved approximations are obtained as follows. Denote the transform variables associated with δ and q by $\gamma \in \mathbb{C}$ and $z \in \mathbb{C}$, respectively. Knowing, that $z = \exp(sT)$ where s is the complex Laplace-operator it is not difficult to see, that

$$s = \frac{1}{T}\ln(1+\gamma T) \tag{3.82}$$

Using a Taylor-series expansion of $\ln(\gamma T + 1)$ yields

$$\begin{aligned} s &= \frac{1}{T}\left(\gamma T - \frac{(\gamma T)^2}{2} + \frac{(\gamma T)^3}{3} - + \cdots\right) && (3.83)\\ &= \gamma - T\frac{\gamma^2}{2} + T^2\frac{\gamma^3}{3} - + \cdots && (3.84) \end{aligned}$$

$$= \sum_{i=1}^{\infty} \frac{(-T)^{i-1}}{i} \gamma^i \tag{3.85}$$

$$= \sum_{i=1}^{\infty} v_{1,i} \gamma^i \tag{3.86}$$

where $v_{1,i},\ i = 1, \ldots, \infty$ are the coefficients of the Taylor-series expansion of s. By raising the above expression in the j'th power it is obvious, that s^j can be expressed in terms of $\gamma^j, \gamma^{j+1}, \ldots$

$$s^j = \left(\sum_{i=1}^{\infty} v_{1,i} \gamma^i \right)^j = \sum_{i=j}^{\infty} v_{j,i} \gamma^i \tag{3.87}$$

where the coefficients $v_{j,i}$ can be determined from $v_{1,1}, \ldots, v_{1,i}$ by repeated convolution of (3.86). By breaking of all these expansions at the N_y'th power of γ we end up with the following matrix formula for approximation of $s, s^2, \ldots, s^{N_y}$

$$\begin{bmatrix} s^0 \\ s \\ s^2 \\ \vdots \\ s^{N_y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & v_{1,1} & v_{1,2} & \cdots & v_{1,N_y} \\ 0 & 0 & v_{2,2} & \cdots & v_{2,N_y} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & v_{N_y,N_y} \end{bmatrix} \begin{bmatrix} \gamma^0 \\ \gamma \\ \gamma^2 \\ \vdots \\ \gamma^{N_y} \end{bmatrix} + \boldsymbol{o}(\gamma^{N_y}) \tag{3.88}$$

or

$$\boldsymbol{S} = \boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{\Gamma} + \boldsymbol{o}(\gamma^{N_y}) \tag{3.89}$$

with obvious definitions of $\boldsymbol{S}$ and $\boldsymbol{\Gamma}$. The matrix $\boldsymbol{M}_{s\gamma}(N_y)$ is upper-triangular with unit diagonal elements and hence of full rank. Combining the time-domain version of the above with (3.79) yields

$$y(t+\tau) = y(t) + \sum_{j=1}^{N_y} \sum_{i=j}^{N_y} \frac{\tau^j}{j!} v_{j,i} \delta^i y(t) + \epsilon \tag{3.90}$$

where ϵ is the approximation error which contains terms with τ of multiplicity greater than N_y and output δ-derivatives of order higher than N_y. Note, that in the above expression t must be a multiple of the sampling period, as we are working in discrete-time. However, with some abuse of notation $y(t+\tau)$ is a continuous approximation of the future output, i.e. τ may be varied continuously. Let $y(t,\tau)$ denote this continuous approximation. Then

$$y(t,\tau) \triangleq y(t) + \sum_{j=1}^{N_y}\sum_{i=j}^{N_y} \frac{\tau^j}{j!} v_{j,i}\delta^i y(t) \tag{3.91}$$

Define the vector $\boldsymbol{y}$ containing the δ-derivatives of $y(t)$

$$\boldsymbol{y} = \begin{bmatrix} y(t) & \delta y(t) & \delta^2 y(t) & \dots & \delta^{N_y} y(t) \end{bmatrix}^\top \tag{3.92}$$

Then in a matrix notation

$$\boxed{y(t,\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{y} \qquad (3.93)}$$

An interesting feature of this improved predictor is shown in the following theorem.

Theorem 10 *Let k denote a positive integer. Then*

$$\boldsymbol{T}_{N_y}(kT)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{\Gamma} = (1+\gamma T)^k \tag{3.94}$$

if the predictor order is $N_y \geq k$. □

Proof: *On noting, that*

$$(1+\gamma T)^k = \exp\left(\ln(1+\gamma T)^k\right) = \exp\left(k\ln\left(1+\gamma T\right)\right) \tag{3.95}$$

we can write $(1+\gamma T)^k$ as the Taylor-series expansion

$$\begin{aligned}(1+\gamma T)^k &= 1 + k\ln(1+\gamma T) + \frac{1}{2!}k^2\ln^2(1+\gamma T) + \cdots \\ &= 1 + kT\left(\frac{1}{T}\ln(1+\gamma T)\right) + \frac{(kT)^2}{2!}\left(\frac{1}{T^2}\ln^2(1+\gamma T)\right) + \cdots\end{aligned}$$

$$
\begin{aligned}
&= 1 + \sum_{i=1}^{\infty} \frac{(kT)^i}{i!} \left(\frac{1}{T^i} \ln^i (1+\gamma T) \right) \\
&= \sum_{i=0}^{\infty} \frac{(kT)^i}{i!} s^i
\end{aligned} \tag{3.96}
$$

where s is given by (3.82). In a matrix notation we have

$$
\begin{aligned}
(1+\gamma T)^k &= \begin{bmatrix} 1 \; kT \; \frac{1}{2!}(kT)^2 \; \cdots \end{bmatrix} \begin{bmatrix} 1 \\ s \\ s^2 \\ \vdots \end{bmatrix} \\
&= \begin{bmatrix} 1 \; kT \; \frac{1}{2!}(kT)^2 \; \cdots \end{bmatrix} \begin{bmatrix} 1 \\ \gamma - T\frac{\gamma^2}{2} + T^2\frac{\gamma^3}{3} - + \cdots \\ \left(\gamma - T\frac{\gamma^2}{2} + T^2\frac{\gamma^3}{3} - + \cdots\right)^2 \\ \vdots \end{bmatrix}
\end{aligned} \tag{3.97}
$$

It is obvious, that only powers of γ of degree less than or equal to k are represented in $(1+\gamma T)^k$. Hence, when making series expansions of s, s^2, $\ldots$ in terms of γ as in (3.87) it suffices to keep powers of γ of order up to k in all these expansions. All higher order terms will cancel each other. We note, that the rows of $\boldsymbol{M}_{s\gamma}$ are exactly the coefficients of these finite series expansions. Furthermore, it appears from (3.87) that powers of s greater than k contain only γ power greater than k and hence, these terms must also be left out. It follows, that with $N_y = k$ we can write

$$
(1+\gamma T)^k = \boldsymbol{T}_{N_y}(kT)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{\Gamma} \tag{3.98}
$$

By increasing N_y we include all terms containing powers of γ of order up to N_y in the series expansions of s, s^2, $\ldots$. However, since $(1+\gamma T)^k$ only contains powers of γ up to order k all these extra terms must cancel each other. This simply means, that the last $N_y - k$ columns of $\boldsymbol{T}_{N_y}(kT)\boldsymbol{M}_{s\gamma}(N_y)$ are zero, i.e. if $N_y \geq k$ then

$$
\boldsymbol{T}_{N_y}(kT)\boldsymbol{M}_{s\gamma}(N_y) = \begin{bmatrix} 1 \; \binom{k}{1} T \; \ldots \; \binom{k}{k} T^k \; 0 \; \ldots \; 0 \end{bmatrix} \tag{3.99}
$$

where the number of zeros is $N_y - k$. Hence, equation (3.98) also holds for $N_y > k$. This completes the proof. ■

The implication of the theorem is as follows. Application to (3.93) gives after a few simple manipulations

$$y(t, kT) = (1 + \delta T)^k y(t) = q^k y(t) = y(t + kT) \tag{3.100}$$

i.e. if perfect knowledge of the δ-derivatives of $y(t)$ is available (the deterministic case), then the predictor is exact at sampling instants when the predictor order is greater than or equal to the desired number of future predicted samples. Hence, in this case no information is wasted. However, the predictor can still be used to predict inter-sample behaviour and to extrapolate by predicting more than N_y samples ahead. This will be exemplified later.

Example 2 *Consider the case $N_y = 4$ and $T = 0.1$. Equation (3.85) gives the following approximation of s*

$$s \approx \gamma - \frac{0.1}{2}\gamma^2 + \frac{(0.1)^2}{3}\gamma^3 - \frac{(0.1)^3}{4}\gamma^4 \tag{3.101}$$

$$= \gamma - 0.05\gamma^2 + 0.003333\gamma^3 - 0.00025\gamma^4 \tag{3.102}$$

The approximation of s^2 is deduced from the following convolution

$$\left(\gamma - 0.05\gamma^2 + 0.003333\gamma^3\right)^2 \tag{3.103}$$

$$= \gamma^2 - 0.1\gamma^3 + 0.009167\gamma^4 - 0.0003333\gamma^5 + 0.00001111\gamma^6 \tag{3.104}$$

Keeping terms up to γ^4 yields

$$s^2 \approx \gamma^2 - 0.1\gamma^3 + 0.009167\gamma^4 \tag{3.105}$$

The approximation of s^3 is deduced from the convolution

$$\left(\gamma^2 - 0.1\gamma^3\right)\left(\gamma - 0.05\gamma^2\right) \tag{3.106}$$

$$= \gamma^3 - 0.15\gamma^4 + 0.005\gamma^5 \tag{3.107}$$

which gives

$$s^3 \approx \gamma^3 - 0.15\gamma^4 \tag{3.108}$$

and finally

$$s^4 \approx \gamma^4 \tag{3.109}$$

Collecting these polynomials in a matrix notation gives

$$\begin{bmatrix} 1 \\ s \\ s^2 \\ s^3 \\ s^4 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -0.05 & 0.003333 & -0.00025 \\ 0 & 0 & 1 & -0.1 & 0.009167 \\ 0 & 0 & 0 & 1 & -0.15 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ \gamma \\ \gamma^2 \\ \gamma^3 \\ \gamma^4 \end{bmatrix} \tag{3.110}$$

□

3.4.2 Calculating $\boldsymbol{M}_{s\gamma}(N_y)$

It is not possible to give a simple explicit expression for the elements of $\boldsymbol{M}_{s\gamma}$. However, in this section a simple recursion formula for the derivatives of subsequent powers of s with respect to γ will be derived. This gives a somewhat more direct way of determining the coefficients of $\boldsymbol{M}_{s\gamma}$.

Consider Leibnitz's rule for high order derivatives of products with $k \geq 1,\ n \geq 1$

$$\begin{aligned} \frac{d^n s^k}{d\gamma^n} &= \frac{d^n}{d\gamma^n}(s s^{k-1}) && (3.111) \\ &= s\frac{d^n s^{k-1}}{d\gamma^n} + \binom{n}{1}\frac{ds}{d\gamma}\frac{d^{n-1}s^{k-1}}{d\gamma^{n-1}} + \cdots + \frac{d^n s}{d\gamma^n}s^{k-1} && (3.112) \\ &= \sum_{i=0}^{n}\binom{n}{i}\frac{d^i s}{d\gamma^i}\frac{d^{n-i}s^{k-1}}{d\gamma^{n-i}} && (3.113) \end{aligned}$$

Let $d_{k,n}$ equal the n'th order derivative of s^k evaluated in $\gamma = 0$, i.e.

$$d_{k,n} = \left. \frac{d^n s^k}{d\gamma^n} \right|_{\gamma=0} \tag{3.114}$$

On noting, that $s(\gamma)\big|_{\gamma=0} = 0$ it follows that

$$d_{k,n} = 0 \qquad \text{for } n < k \tag{3.115}$$

Evaluating (3.113) for $\gamma = 0$ then immediately gives the recursion

$$d_{k,n} = \sum_{i=1}^{n-k+1} \binom{n}{i} d_{1,i} d_{k-1,n-i} \qquad k > 1 \ , \ n > 1 \tag{3.116}$$

The recursion is initialized by calculating

$$d_{1,i} = (-T)^{i-1}(i-1)! \quad \text{for } i = 1, \ldots, N_y \tag{3.117}$$

which can be seen from e.g. (3.85). Define the upper-triangular matrix

$$\boldsymbol{D}(N_y) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & d_{1,1} & d_{1,2} & \cdots & d_{1,N_y} \\ 0 & 0 & d_{2,2} & \cdots & d_{2,N_y} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & d_{N_y,N_y} \end{bmatrix} \tag{3.118}$$

The recursion formula (3.116) specifies, that for $k > 1$ row k can be obtained from the elements of row $k-1$ and $d_{1,i}$, $i = 1, \ldots, N_y - k + 1$, i.e. the second row.

To get the matrix $\boldsymbol{M}_{s\gamma}$ we still need to multiply the derivatives with the corresponding inverse factorials in the Taylor-series expansion. This can be achieved by defining a diagonal matrix containing these factors, i.e. define

$$\boldsymbol{F}(N_y) = \begin{bmatrix} \frac{1}{0!} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{1!} & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{2!} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{(N_y)!} \end{bmatrix} \tag{3.119}$$

Then it readily follows, that

$$\boldsymbol{M}_{s\gamma}(N_y) = \boldsymbol{D}(N_y)\boldsymbol{F}(N_y) \tag{3.120}$$

These formulas provide a more obvious connection between the coefficients of $\boldsymbol{M}_{s\gamma}$. However, in terms of computation time a direct convolution as suggested earlier is more efficient. Using the convolution technique the approximation of s^k can in general be deduced from a convolution of the approximation of s^{k-1} keeping terms up to an order of $N_y - 1$ and the approximation of s keeping terms up to order $N_y + 1 - k$. The number of non-zero coefficients in both these polynomials decreases as k increases and hence the additional computational effort involved in determining the approximation of s^k given the approximation of s^{k-1} decreases as k increases, see also Example 2.

3.4.3 Emulation of output δ-derivatives

The δ-derivatives in (3.93) are unknown at time t (the δ-operator is non-causal) and must therefore be estimated. This is achieved by a δ-domain version of the time-derivative emulation method presented in (Demircioğlu and Gawthrop 1991). The term *emulator* is introduced as a much more general concept in (Gawthrop 1987) than as considered here. Emulators are used to emulate physically unrealizable operations such as derivation, cancellation of non-minimumphase zeros and realizing inverse time-delays. Here we use the term δ-domain emulator to denote an estimation of δ-derivatives of the system output.

Consider again the δ-domain ARMAX model

$$A_\delta(\delta)y(t) = B_\delta(\delta)u(t) + C_\delta(\delta)e(t) \tag{3.121}$$

where $A_\delta(\delta)$ and $C_\delta(\delta)$ are assumed monic and

$$\begin{aligned} \deg(A_\delta) &= n & (3.122)\\ \deg(B_\delta) &= m < n & (3.123)\\ \deg(C_\delta) &= n - \Delta & (3.124) \end{aligned}$$

where $\Delta \in \{0, 1\}$, i.e. $C_\delta(\delta)/A_\delta(\delta)$ is either exactly proper or strictly proper. Choosing $\deg(C_\delta) = \deg(A_\delta) = n$ so that $\Delta = 0$ makes it possible to model white noise acting directly on the system output. This is not possible with $\Delta = 1$. Again, $C_\delta(\delta)$ is assumed to have all its roots strictly inside the δ-domain stability circle. Introduce the notation

$$y_k(t) \triangleq \delta^k y(t) \tag{3.125}$$

Consider this k'th order δ-derivative expressed by means of the system equation

$$y_k(t) = \delta^k \frac{B_\delta}{A_\delta} u(t) + \delta^k \frac{C_\delta}{A_\delta} e(t) \tag{3.126}$$

The noise term can be split up into a term consisting of future noise values and a term consisting of past noise values be means of the Diophantine-equation

$$\delta^k \frac{C_\delta(\delta)}{A_\delta(\delta)} = E_e^{[k]}(\delta) + \frac{F_e^{[k]}(\delta)}{A_\delta(\delta)} \tag{3.127}$$

where $F_e^{[k]}/A_\delta$ is strictly proper. The orders of $E_e^{[k]}$ and $F_e^{[k]}$ are restricted by

$$\begin{aligned} \deg(E_e^{[k]}) &= \max\{k - \Delta, 0\} & (3.128)\\ \deg(F_e^{[k]}) &= n - 1 & (3.129) \end{aligned}$$

This gives

$$y_k(t) = \delta^k \frac{B_\delta}{A_\delta} u(t) + \frac{F_e^{[k]}}{A_\delta} e(t) + E_e^{[k]} e(t) \tag{3.130}$$

Note, that the term $E_e^{[k]}(\delta)e(t)$ depends on the noise signals $e(t), e(t + T), \ldots, e(t + (k - \Delta)T)$. Hence, according to Table 3.1 this term is exactly unknown only in modelling cases $\mathcal{M}_1$ and $\mathcal{M}_2$. This motivates the special interest in these two cases that has been shown throughout the chapter. It is not possible to treat the case of complete information with $\Delta = 0$ ($\mathcal{M}_0$) or the case of incomplete information with $\Delta = 1$ ($\mathcal{M}_3$) without introducing a factor $q = 1 + \delta T$ in the calculations, as indicated in (3.26), see also Section 3.2.6. This will ruin the close connection to the continuous-time case because the emulators will then no longer have a well-defined limit as $T \to 0$.

The past noise values are not directly available from $\mathcal{I}_{\Delta,t,k}$ but can be reconstructed from measurable data, i.e.

$$\frac{F_e^{[k]}}{A_\delta} e(t) = \frac{F_e^{[k]}}{A_\delta C_\delta} \left(A_\delta y(t) - B_\delta u(t)\right) \tag{3.131}$$

Insertion into (3.130) yields

$$y_k(t) = \frac{B_\delta}{A_\delta} \left(\delta^k - \frac{F_e^{[k]}}{C_\delta}\right) u(t) + \frac{F_e^{[k]}}{C_\delta} y(t) + E_e^{[k]} e(t) \tag{3.132}$$

Using (3.127) gives

$$y_k(t) = \frac{B_\delta E_e^{[k]}}{C_\delta} u(t) + \frac{F_e^{[k]}}{C_\delta} y(t) + E_e^{[k]} e(t) \tag{3.133}$$

Next the effect of future and past control input is separated by means of the Diophantine-equation

$$\frac{E_e^{[k]}(\delta) B_\delta(\delta)}{C_\delta(\delta)} = G_e^{[k]}(\delta) + \frac{H_e^{[k]}(\delta)}{C_\delta(\delta)} \tag{3.134}$$

with order restrictions

$$\begin{aligned}\deg(G_e^{[k]}) &= \max\{k-d, 0\} \qquad (3.135)\\ \deg(H_e^{[k]}) &= \min\{n-1-\Delta, m+\deg(E_e^{[k]})\} \qquad (3.136)\end{aligned}$$

where $H_e^{[k]}/C_\delta$ is strictly proper. We can then finally write

$$\boxed{y_k(t) = \underbrace{G_e^{[k]}u(t)}_{\text{future}} + \underbrace{\frac{H_e^{[k]}}{C_\delta}u(t) + \frac{F_e^{[k]}}{C_\delta}y(t)}_{\text{known}} + \underbrace{E_e^{[k]}e(t)}_{\text{future}} \qquad (3.137)}$$

Note, that in case $\Delta = 0$ both $H_e^{[k]}/C_\delta$ and $F_e^{[k]}/C_\delta$ are strictly proper so that the two terms including these transfer functions in (3.137) represent the effect of past control and output signals. In case $\Delta = 1$ the transfer function $F_e^{[k]}/C_\delta$ becomes exactly proper which means that also the present output signal $y(t)$ is taken into account (complete information is utilized).

Example 3 *Consider the δ-domain description*

$$\left(\delta^2+\delta+1\right)y(t) = (0.1\delta+1)\,u(t) + \left(\delta^2+8\delta+15\right)e(t) \qquad (3.138)$$

Let $k=3$. The solutions to the Diophantine equation (3.127) become

$$\begin{aligned}E_e^{[0]}(\delta) &= 1 & F_e^{[0]}(\delta) &= 7\delta+14\\ E_e^{[1]}(\delta) &= \delta+7 & F_e^{[1]}(\delta) &= 7\delta-7\\ E_e^{[2]}(\delta) &= \delta^2+7\delta+7 & F_e^{[2]}(\delta) &= -14\delta-7\\ E_e^{[3]}(\delta) &= \delta^3+7\delta^2+7\delta-14 & F_e^{[3]}(\delta) &= 7\delta+14\end{aligned} \qquad (3.139)$$

and the solutions to (3.134) become

$$\begin{aligned}G_e^{[0]}(\delta) &= 0 & H_e^{[0]}(\delta) &= 0.1\delta+1\\ G_e^{[1]}(\delta) &= 0.1 & H_e^{[1]}(\delta) &= 0.9\delta+5.5\\ G_e^{[2]}(\delta) &= 0.1\delta+0.9 & H_e^{[2]}(\delta) &= -1\delta-6.5\\ G_e^{[3]}(\delta) &= 0.1\delta^2+0.9\delta-1 & H_e^{[3]}(\delta) &= 0.1\delta+1\end{aligned} \qquad (3.140)$$

The output δ-derivative of order 3 can thus be written as

$$\begin{aligned} y_3(t) &= \left(0.1\delta^2 + 0.9\delta - 1\right) u(t) + \frac{0.1\delta + 1}{\delta^2 + 8\delta + 15} u(t) \\ &\quad + \frac{7\delta + 14}{\delta^2 + 8\delta + 15} y(t) + \left(\delta^3 + 7\delta^2 + 7\delta - 14\right) e(t) \end{aligned} \tag{3.141}$$

If we reduce the order of $C_\delta(\delta)$ by choosing

$$C_\delta(\delta) = \delta + 5 \tag{3.142}$$

so that $\Delta = 1$ then the solutions to the Diophantine equations become

$$\begin{array}{rclcrcl} E_e^{[0]}(\delta) &=& 0 & & F_e^{[0]}(\delta) &=& \delta + 5 \\ E_e^{[1]}(\delta) &=& 1 & & F_e^{[1]}(\delta) &=& 4\delta - 1 \\ E_e^{[2]}(\delta) &=& \delta + 4 & & F_e^{[2]}(\delta) &=& -5\delta - 4 \\ E_e^{[3]}(\delta) &=& \delta^2 + 4\delta - 5 & & F_e^{[3]}(\delta) &=& \delta + 5 \end{array} \tag{3.143}$$

and

$$\begin{array}{rclcrcl} G_e^{[0]}(\delta) &=& 0 & & H_e^{[0]}(\delta) &=& 0 \\ G_e^{[1]}(\delta) &=& 0.1 & & H_e^{[1]}(\delta) &=& 0.5 \\ G_e^{[2]}(\delta) &=& 0.1\delta + 0.9 & & H_e^{[2]}(\delta) &=& -0.5 \\ G_e^{[3]}(\delta) &=& 0.1\delta^2 + 0.9\delta - 1 & & H_e^{[3]}(\delta) &=& 0 \end{array} \tag{3.144}$$

Note, that the G_e polynomials are not affected by the change in $C_\delta(\delta)$. This is true in general. The output δ-derivative of order 3 can in this case be written as

$$y_3(t) = \left(0.1\delta^2 + 0.9\delta - 1\right) u(t) + y(t) + \left(\delta^2 + 4\delta - 5\right) e(t) \tag{3.145}$$

□

The emulation of $y_k(t)$ is obtained by taking expectation conditioned on $\mathcal{I}_{\Delta,t,k}$. This gives the emulation

$$y_k^*(t) \triangleq \mathcal{E}\left\{y_k(t)|\mathcal{I}_{\Delta,t,k}\right\} = G_e^{[k]} u(t) + y_k^0(t) \tag{3.146}$$

where $\mathcal{E}$ is mathematical expectation and

$$y_k^0(t) = \frac{H_e^{[k]}}{C_\delta} u(t) + \frac{F_e^{[k]}}{C_\delta} y(t) \tag{3.147}$$

The signal $y_k^0(t)$ can be interpreted as a *free emulation*, i.e. with no future input $y_k^0(t)$ describes the evolution of the δ-derivatives of the output.

In a matrix notation we can write

$$\boldsymbol{y}^* = \tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}} + \boldsymbol{y}^0 \tag{3.148}$$

where

$$\boldsymbol{y}^* = \left[y_0^*(t) \; y_1^*(t) \; \dots \; y_{N_y}^*(t) \right]^\top \tag{3.149}$$

$$\boldsymbol{y}^0 = \left[y_0^0(t) \; y_1^0(t) \; \dots \; y_{N_y}^0(t) \right]^\top \tag{3.150}$$

$$\tilde{\boldsymbol{u}} = \left[u(t) \; \delta u(t) \; \dots \; \delta^{(N_y-d)} u(t) \right]^\top \tag{3.151}$$

so that the dimension of $\tilde{\boldsymbol{G}}$ is

$$\dim(\tilde{\boldsymbol{G}}) = (N_y + 1) \times (N_y - d + 1) \tag{3.152}$$

Row k of $\tilde{\boldsymbol{G}}$ is built up from the coefficients of the G_e polynomials in the following manner (illustrated for d=1)

$$\tilde{\boldsymbol{G}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ g_{e,0}^{[1]} & 0 & \cdots & 0 \\ g_{e,0}^{[2]} & g_{e,1}^{[2]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{e,0}^{[N_y]} & g_{e,1}^{[N_y]} & \cdots & g_{e,N_y-1}^{[N_y]} \end{bmatrix} \tag{3.153}$$

where (for $d = 1$)

$$G_e^{[k]}(\delta) = g_{e,k-1}^{[k]} \delta^{k-1} + \cdots + g_{e,1}^{[k]} \delta + g_{e,0}^{[k]} \tag{3.154}$$

We can now finally write the emulator-based predictor as

$$\hat{y}^*(t+\tau) \triangleq \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{y}^* \tag{3.155}$$

or with (3.148) inserted

$$\boxed{\hat{y}^*(t+\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\left(\tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}} + \boldsymbol{y}^0\right) \qquad (3.156)}$$

The above expressions have been obtained from the predictor equation (3.93) simply by replacing the vector $\boldsymbol{y}$ containing the true but unknown δ-derivatives with the vector $\boldsymbol{y}^*$ containing estimates of the true δ-derivatives.

From the derivation of the above emulator-based predictor it appears, that no problems occur in the continuous-time limit $T \to 0$. Actually, for fixed N_y the emulator-based predictor has a well-defined limit as $T \to 0$. It converges to a predictor similar to the continuous-time T-ahead predictor presented in (Demircioğlu and Gawthrop 1991). This is discussed in more detail in Section 3.4.4

Note, that the Diophantine equations (3.127) and (3.134) are equivalent to the Diophantine equation (3.13) and (3.20) of the q-domain optimal predictor if the symbol δ is interchanged with the symbol q. With this interpretation the algorithm shown in Table 3.2 can be used to solve (3.127) and (3.134) recursively with no modifications other than name changes. The system polynomials A_q, B_q and C_q should be replaced by A_δ, B_δ and C_δ, E_q^j with E_e^j and so forth, see Table 3.4.

The update of G_e^j shows, that the coefficients shift from one iteration to the next, i.e.

$$g_{e,k-1}^{[k]} = g_{e,k-2}^{[k-1]} \tag{3.157}$$

Because of this shifting structure we can actually express the matrix $\tilde{\boldsymbol{G}}$ in

Table 3.4: *Algorithm to recursively solve Diophantine-equations for emulator-based predictor in case of complete($\Delta = 1$)/incomplete($\Delta = 0$) information.*

- *Initialization*

  1. $E_e^{[0]} = \begin{cases} 1 & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

  2. $F_e^{[0]} = \begin{cases} C_\delta - A_\delta & \text{if } \Delta = 0 \\ C_\delta & \text{if } \Delta = 1 \end{cases}$

  3. $G_e^{[0]} = \begin{cases} 0 & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

  4. $H_e^{[0]} = \begin{cases} B_\delta & \text{if } \Delta = 0 \\ 0 & \text{if } \Delta = 1 \end{cases}$

- *Recursion*

  1. $\alpha = f_{e,n-1}^{[j]}$

  2. $E_e^{[j+1]} = \delta E_e^{[j]} + \alpha$

  3. $F_e^{[j+1]} = \delta F_e^{[j]} - \alpha A_\delta$

  4. $\beta = \begin{cases} h_{e,n-1}^{[j]} & \text{if } \Delta = 0 \\ h_{e,n-2}^{[j]} + \alpha b_{\delta,n-1} & \text{if } \Delta = 1 \end{cases}$

  5. $G_e^{[j+1]} = \delta G_e^{[j]} + \beta$

  6. $H_e^{[j+1]} = \delta H_e^{[j]} - \beta C_\delta + \alpha B_\delta$

terms of only $G_e^{[N_y]}$ as illustrated below (again for $d=1$)

$$\tilde{\boldsymbol{G}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ g_{e,N_y-1}^{[N_y]} & 0 & \cdots & 0 \\ g_{e,N_y-2}^{[N_y]} & g_{e,N_y-1}^{[N_y]} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{e,0}^{[N_y]} & g_{e,1}^{[N_y]} & \cdots & g_{e,N_y-1}^{[N_y]} \end{bmatrix} \tag{3.158}$$

We remark, that $\tilde{\boldsymbol{G}}$ is Toeplitz, i.e. constant along the diagonals.

3.4.4 Connection to continuous-time

In Chapter 2 we have seen, that δ-domain models converge to their continuous-time equivalents as the sampling period tends to zero. This means, that the Diophantine-equations involved in finding the emulations of output δ-derivatives become algebraically equivalent to the ones leading to the time-derivative emulators derived in (Demircioğlu and Gawthrop 1991). In other words, assume that the δ-domain ARMAX model converge to a continuous-time model of the form

$$A_s(s)Y(s) = B_s(s)U(s) + C_s(s)E(s) \tag{3.159}$$

where $Y(s)$, $U(s)$ and $E(s)$ is the transformed system output, input and disturbance input, respectively, and $A_s(s)$, $B_s(s)$ and $C_s(s)$ are polynomials in the Laplace-operator. Furthermore, assume zero initial conditions. Then the frequency-domain continuous-time version of (3.137) becomes

$$s^k Y(s) = G_s^{[k]}(s)U(s) + \frac{H_s^{[k]}(s)}{C_s(s)}U(s) + \frac{F_s^{[k]}(s)}{C_s(s)}Y(s) + E_s^{[k]}(s)E(s) \tag{3.160}$$

where the polynomials are found by solving the Diophantine-equations

$$s^k \frac{C_s(s)}{A_s(s)} = E_s^{[k]}(s) + \frac{F_s^{[k]}(s)}{A_s(s)} \tag{3.161}$$

$$\frac{E_s^{[k]}(s)B_s(s)}{C_s(s)} = G_s^{[k]}(s) + \frac{H_s^{[k]}(s)}{C_s(s)} \tag{3.162}$$

The time-derivative emulation corresponding to (3.146) is obtained as

$$Y_k^*(s) = G_s^{[k]}(s)U(s) + \frac{H_s^{[k]}(s)}{C_s(s)}U(s) + \frac{F_s^{[k]}(s)}{C_s(s)}Y(s) \tag{3.163}$$

with corresponding error $E_k^*(s) = E_s^{[k]}(s)E(s)$. Denote as in the δ-domain, the corresponding time-domain emulator by $y_k^*(t)$. Then on noting, that

$$\lim_{T\to 0} \boldsymbol{M}_{s\gamma}(N_y) = \boldsymbol{I} \tag{3.164}$$

the continuous-time output approximation becomes

$$\hat{y}^*(t+\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{y}^* \tag{3.165}$$

exactly as in (Demircioğlu and Gawthrop 1991).

The statistical interpretation of the δ-derivative emulators as the conditional expectation of the output δ-derivative given available information, see (3.146), is lost in the continuous-time case. Apart from this, as illustrated above, the emulator-based predictor is well-defined in the continuous-time limit.

3.4.5 Connection to optimal prediction

An interesting connection between optimal prediction and emulator-based prediction is established in the following theorem.

Theorem 11 *Let k denote a positive integer. Then*

$$\hat{y}^*(t+kT) = \hat{y}(t+kT|\mathcal{I}_{\Delta,t,k}) \tag{3.166}$$

if the predictor order is $N_y \geq k$. □

Proof: *From the binomial formula we immediately get*

$$y(t+kT) = \sum_{i=0}^{k} T^i \binom{k}{i} \delta^i y(t) = \sum_{i=0}^{k} T^i \binom{k}{i} y_i(t) \tag{3.167}$$

Taking conditional expectation in (3.167) shows, according to (3.146) and utilizing the linearity of the conditional expectation operator, that

$$\hat{y}(t+kT|\mathcal{I}_{\Delta,t,k}) = \sum_{i=0}^{k} T^i \binom{k}{i} y_i^*(t) \tag{3.168}$$

From Theorem 10, see (3.99), it now appears, that if $N_y \geq k$ then

$$\hat{y}(t+kT|\mathcal{I}_{\Delta,t,k}) = \boldsymbol{T}_{N_y}(kT)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{y}^* = \hat{y}^*(t+kT) \tag{3.169}$$

■

The theorem shows that the emulator-based predictor contains the optimal predictor as a subcase. The optimal predictions are obtained at the first N_y sampling instants following time t. Hence, the theorem establishes a bridge between the continuous-time T-ahead predictor presented in (Demircioğlu and Gawthrop 1991) for continuous-time models and optimal minimum-variance predictions based on discrete-time δ-domain models. The T-ahead predictor is obtained in the limit $T \to 0$ and the minimum-variance predictor is obtained at the sampling instants if N_y is sufficiently large.

A direct connection between the polynomials of the Diophantine-equations in the emulator-based and the optimal case can also easily be made. Insertion of (3.137) into (3.167) yields

$$\begin{aligned} y(t+kT) \quad = \quad & \sum_{i=0}^{k} T^i \binom{k}{i} \left(G_e^{[i]}(\delta)u(t) + \frac{H_e^{[i]}(\delta)}{C_\delta(\delta)} u(t) \right. \\ & \left. + \frac{F_e^{[i]}(\delta)}{C_\delta(\delta)} y(t) + E_e^{[i]}(\delta)e(t) \right) \end{aligned} \tag{3.170}$$

In the treatment of optimal prediction we found (3.77)

$$y(t+kT) = G_\delta^{[k]}u(t) + \frac{H_\delta^{[k]}}{C_\delta}u(t) + \frac{F_\delta^{[k]}}{C_\delta}y(t) + E_\delta^{[k]}e(t) \tag{3.171}$$

Comparison of these expressions shows the following relations

$$G_\delta^{[k]}(\delta) = \sum_{i=0}^{k} T^i \binom{k}{i} G_e^{[i]}(\delta) \tag{3.172}$$

$$H_\delta^{[k]}(\delta) = \sum_{i=0}^{k} T^i \binom{k}{i} H_e^{[i]}(\delta) \tag{3.173}$$

$$F_\delta^{[k]}(\delta) = \sum_{i=0}^{k} T^i \binom{k}{i} F_e^{[i]}(\delta) \tag{3.174}$$

$$E_\delta^{[k]}(\delta) = \sum_{i=0}^{k} T^i \binom{k}{i} E_e^{[i]}(\delta) \tag{3.175}$$

Similarly, from

$$\delta^k y(t) = \left(\frac{q-1}{T}\right)^k y(t) = T^{-k}\sum_{i=0}^{k}(-1)^{i-k}\binom{k}{i} q^i y(t) \tag{3.176}$$

we find

$$\delta^k y(t) = T^{-k}\sum_{i=0}^{k}(-1)^{i-k}\binom{k}{i}\left(G_\delta^{[i]}u(t) + \frac{H_\delta^{[i]}}{C_\delta}u(t) + \frac{F_\delta^{[i]}}{C_\delta}y(t) + E_\delta^{[i]}e(t)\right) \tag{3.177}$$

which gives the inverse relations

$$G_e^{[k]}(\delta) = T^{-k}\sum_{i=0}^{k}(-1)^{i-k}\binom{k}{i} G_\delta^{[i]}(\delta) \tag{3.178}$$

$$H_e^{[k]}(\delta) = T^{-k}\sum_{i=0}^{k}(-1)^{i-k}\binom{k}{i} H_\delta^{[i]}(\delta) \tag{3.179}$$

$$F_e^{[k]}(\delta) = T^{-k}\sum_{i=0}^{k}(-1)^{i-k}\binom{k}{i} F_\delta^{[i]}(\delta) \tag{3.180}$$

$$E_e^{[k]}(\delta) \quad = \quad T^{-k}\sum_{i=0}^{k}(-1)^{i-k}\binom{k}{i}E_\delta^{[i]}(\delta) \tag{3.181}$$

3.5 Simulation examples

In this section we illustrate δ-domain emulator-based prediction by simple simulation examples.

Example 4 *As mentioned earlier the emulator-based predictor has an ability to extrapolate, i.e. to predict longer than N_y samples ahead. This may give computational savings at fast sampling rates which is welcome in long range predictive control schemes. To illustrate the extrapolation facility of the new δ-domain emulator-based predictor some experiments have been performed on the third order stable system*

$$\frac{B(s)}{A(s)} = \frac{3}{(s+3)(s^2+s+1)} \tag{3.182}$$

As argued in (Demircioğlu and Gawthrop 1991) the precision of the predictor can be examined by comparing the true and approximate (predicted) step-responses of the open-loop system. In Figure 3.1 these responses are depicted for various predictor orders $5 \leq N_y \leq 40$ over the time-range $0 \leq t \leq 10$ with sampling periods of $T = 0.1$ and $T = 0.01$.

Clearly, the emulator-based predictor is in both cases capable of predicting the output much further than the "guaranteed" horizon N_yT, cf. Theorem 11. We note, that perhaps somewhat surprising the effective prediction horizon decreases when T is reduced. A possible explanation is, that the N_y δ-derivatives contains information of the output in a longer absolute time range when T is large. □

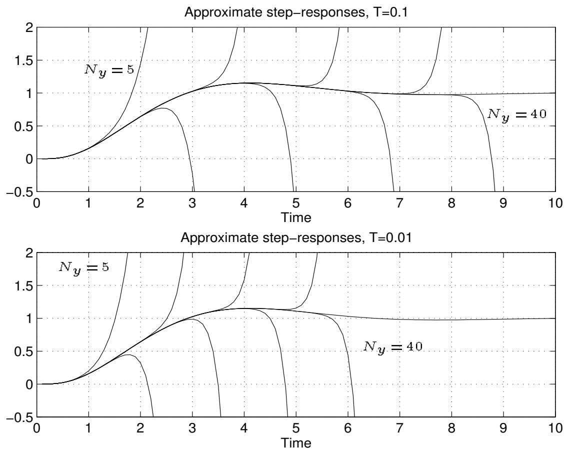

Figure 3.1: True and approximate step-responses with third order system minimum-phase system and varying $N_y = 5, \ldots, 40$ in steps of 5, $T = 0.1$ (top) and $T = 0.01$ (bottom).

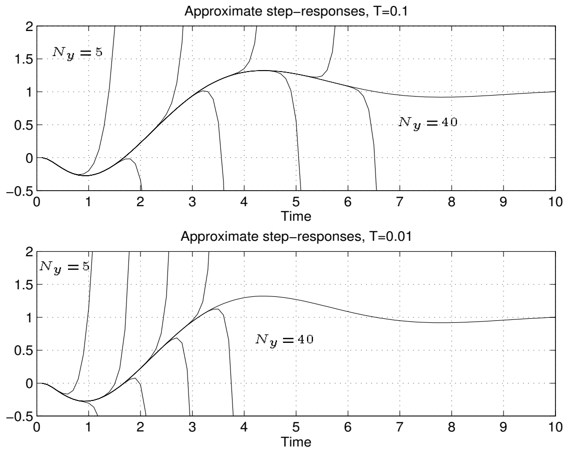

Figure 3.2: True and approximate step-responses with third order system non-minimum phase system and varying $N_y = 5, \ldots, 40$ in steps of 5, $T = 0.1$ (top) and $T = 0.01$ (bottom).

Example 5 *Next we consider the slightly less well-behaved system*

$$\frac{B(s)}{A(s)} = 5\frac{-s+1}{(s+5)(s^2+0.8s+1)} \tag{3.183}$$

which is non-minimum phase and poorly damped. The true and approximate step-responses with similar simulation conditions as above are shown in Figure 3.2.

The results are very similar to the ones obtained in the first simulation example. We notice, that the effective prediction horizon has been somewhat reduced. □

3.6 Chapter conclusion

Three approaches to prediction using discrete-time models have been considered. Firstly, the traditional approach to optimal prediction using discrete-time models formulated in the shift-operator q is described. This approach is then translated to be based on models formulated in the discrete δ-operator. The computation of output predictions requires solution of two coupled Diophantine-equations. Recursive algorithms to solve these equations are given.

Next a new more general prediction approach based on ideas from continuous-time is presented. The output can be predicted continuously even though the predictor is based on discrete-time δ-domain models. The new so-called emulator-based approach is shown to contain the optimal minimum variance predictor as a special case. In addition it possesses an ability to predict the output continuously and even further ahead than the guaranteed horizon where it coincides with the optimal predictor at the sampling instants. This has been illustrated by simulation examples. Due to the properties of δ-domain models the emulator-based predictor has a well-defined limit in the continuous-time case. Moreover, with the connection to the discrete-time optimal predictor in mind this new prediction method provide further links between discrete-time and continuous-time theory. This can be utilized in the construction of predictive control methods as we shall see in the next chapter.

Chapter 4

Predictive Control in the Delta-Domain

4.1 Introduction

In recent years, there have been a lot of research interest in so-called predictive control algorithms. This chapter is devoted to such algorithms with special attention to new formulations in the δ-domain. A unified framework for the treatment of continuous-time and discrete-time predictive control algorithms is developed.

First an introduction to predictive control is given. The popular Generalized Predictive Control (GPC) algorithm is treated in some detail. Then we propose new predictive control algorithms formulated in the δ-domain and using the emulator-based predictor presented in the previous chapter. Several extensions to the basic schemes are presented. This includes frequency weighted criterion functions and both SISO and MIMO state-space formulations.

4.2 Predictive control

We consider so-called model-based predictive control (MPC) algorithms. In MPC a process model is used explicitly to predict the effect of future input on the system output over a given prediction horizon. The control input is then determined by minimizing a desired control criterion. A large number of MPC algorithms have been presented in the literature. They differ in the choice of model structure and criterion function but the methodology is the same. Surveys can be found in (De Keyser *et al.* 1988, García *et al.* 1989, Soeterboek 1992).

In the following we describe the popular Generalized Predictive Controller (GPC) and at the same time introduce some of the basic concepts of predictive control. After that, a survey of recent contributions on extensions to the basic GPC algorithm is given.

4.2.1 Generalized predictive control

In (Åström 1970) the minimum variance controller was presented. This controller minimizes the variance of the d-step-ahead prediction of the control error according to the criterion

$$J_{MV} = \mathcal{E}\left\{(y(t+dT) - w(t+dT))^2\right\} \tag{4.1}$$

where T is the sampling period, d is the time-delay of the plant in samples, $y(t+dT)$ is the system output and $w(t+dT)$ is a reference signal. At time t the criterion is minimized giving an optimal control input $u(t)$. A new problem is solved at time $t+T$. A well-known problem with minimum variance control is the fact, that the strategy only works with minimum-phase system and is very demanding on the power of the input signal.

One way to make the control strategy work also for non-minimum phase systems is to introduce a weighting of the control input

$$J_{GMV} = \mathcal{E}\left\{(y(t+dT) - w(t+dT))^2 + \rho u^2(t)\right\} \tag{4.2}$$

where ρ is positive weighting factor. The control weight also gives a less active control input. This strategy is known as Generalized Minimum Variance control (GMV).

The above algorithms are very sensitive to the estimate of the time-delay of the plant. If the time-delay is poorly known, the algorithms may behave very badly. An algorithm which also copes with unknown time-delay was suggested by Clarke, Mohtadi and Tuffs in (Clarke *et al.* 1987a, Clarke *et al.* 1987b)—the Generalized Predictive Control (GPC) algorithm.

In this section we will describe this algorithm in some detail. The GPC handles many difficult situations in one algorithm without the need for specific modifications. As listed in (Clarke *et al.* 1987a) the GPC can deal with

- Non-minimumphase plants. This is a common problem in discrete-time control as mentioned in Chapter 2: At fast enough sampling rates and with a continuous-time transfer function of relative degree greater than two, the discrete-time description will always contain non-minimumphase sampling zeros.
- Open-loop unstable or badly damped plants.
- Unknown or varying time-delay. Many traditional self-tuning algorithms such as minimum-variance control are very sensitive to the dead-time estimate.

- Plant order overestimation. The GPC handles pole/zero cancellations without problems.

Because of these features the GPC has become very popular and many recent predictive control algorithms originate from the basic GPC algorithm described here.

Criterion function

The criterion function of the GPC is given by

$$J_{GPC} = \mathcal{E}\left\{ \sum_{i=N_1}^{N_2} [y(t+iT) - w(t+iT)]^2 + \rho \sum_{i=0}^{N_2-1} [u(t+iT) - u(t+(i-1)T)]^2 \middle| \mathcal{I}_{c,t} \right\} \quad (4.3)$$

The minimization is subject to the following assumption on future control increments

$$u(t+kT) - u(t+(k-1)T) = 0, \qquad k > N_u \quad (4.4)$$

which is known as a constant future control constraint. It is easy to show, that minimization of the above criterion function is equivalent to minimization of

$$J'_{GPC} = \sum_{i=N_1}^{N_2} [\hat{y}(t+iT|\mathcal{I}_{c,t}) - w(t+iT)]^2 + \rho \sum_{i=0}^{N_u} [u(t+iT) - u(t+(i-1)T)]^2 \quad (4.5)$$

where $\hat{y}(t+iT|\mathcal{I}_{c,t})$ is the i-step ahead minimum-variance prediction of the output given complete information.

As it appears from the criterion function the GPC minimizes the squared difference of the output prediction $\hat{y}(t + iT|\mathcal{I}_{c,t})$ to the reference signal $w(t + iT)$ over a finite prediction horizon. It is possible using ρ to include a weighting of the control increments in the criterion. Weighting increments rather than absolute values is a way of avoiding steady-state errors for systems with no integration.

We identify the following design variables

- Minimum costing horizon N_1
- Maximum costing horizon N_2
- Control horizon N_u
- Control weighting factor ρ

Guidelines for the choice of these design variables are given in (Clarke *et al.* 1987a) and (Soeterboek 1992) where a comprehensive simulation study supports the suggestions.

System model

The basic GPC algorithm is based on an ARIMAX model of the form

$$A(q^{-1})y(t) = q^{-1}B(q^{-1})u(t) + \frac{C(q^{-1})}{1 - q^{-1}}\xi(t) \tag{4.6}$$

where $y(t)$ is the output, $u(t)$ is the input and $\{\xi(t)\}$ is a white noise sequence. Notice, that the noise part of this description is not stationary. The reason for choosing such a description is to enforce integral action in the controller in order to cope with for instance load-disturbances, which are common in industrial applications. By choosing the ARIMAX model structure the integral action is a consequence of the model rather than an *ad hoc* extension to the controller. Also note, that the plant delay is assumed to be equal to the sampling period.

The system (4.6) can equivalently by represented by the ARMAX model description

$$A(q^{-1})(1-q^{-1})y(t) = B(q^{-1})(1-q^{-1})u(t-T) + C(q^{-1})\xi(t) \qquad (4.7)$$

where T is the sampling period.

Often the noise polynomial $C(q^{-1})$ is considered a design variable and is then called the observer polynomial $T(q^{-1})$. This polynomial affects the disturbance rejection and robustness towards unmodelled dynamics of the controller. The role of the observer polynomial is analyzed in (Robinson and Clarke 1991, Mohtadi 1988, Ordys 1993).

The control horizon

An important element of the GPC is the control horizon N_u. Control signals up to time $t+(N_2-1)T$ have effect on the output up to time $t+N_2T$ (with a time-delay of one sample). Rather than allowing all these control signals to be free it is assumed, that after an interval N_u the control increments are zero. This has an effect on the complexity of the minimization problem as the number of free optimization variables is reduced to N_u+1. In particular, with $N_u=0$ there is only one free control signal, $u(t)$. This choice is computationally attractive, especially in the adaptive case.

As discussed in (Clarke *et al.* 1987a) the control horizon N_u is an important design parameter. Generally speaking, the more complex the plant the higher value of N_u is needed to get acceptable control. For many simple processes the choice $N_u=0$ is adequate.

The future reference

An important feature of predictive control algorithms is the possibility to incorporate trajectory planning in the control design. In the calculation of the current control input it is possible to utilize information about the future reference over the prediction horizon. In some applications as e.g. robot control or computer controlled machines it may be useful to take advantage of this knowledge.

If the future reference is not known a sensible reference trajectory must be generated. Several options exist.

1. Predict the reference signal: Based on some model of the reference signal it may be predicted.
2. Generate a smoothed trajectory to the setpoint by means of a reference model.
3. Make direct assumptions on the future reference. A common assumption is that the future reference is constant and equal to the current setpoint.

In the following we will not treat the trajectory planning problem. We either assume, that the future reference is known or use the constant future reference assumption.

The receding horizon principle

The GPC controller and most other predictive controllers are implemented according to the so-called *receding horizon principle*. At each sampling instant t the following is done:

1. Calculate the future reference sequence.

2. Find a set of predicted outputs expressed by means of the unknown future input sequence. That is, find $\hat{y}(t+N_1T), \ldots, \hat{y}(t+N_2T)$ as a function of past data and the future control signal sequence $u(t)$, $u(t+T), \ldots, u(t+N_uT)$.

3. By minimizing the control criterion find an optimal sequence of future inputs; the vector of *projected inputs.*

4. Apply the first element $u(t)$ of this vector to the plant.

Using the receding horizon principle can be seen as a way of correcting the output predictions to future disturbances or modelling errors, see (Soeterboek 1992). It makes it possible to take into account new information obtained at each sample, primarily by measuring the system output. Other information such as known deterministic disturbances, known changes in the setpoint or hard constraints on the input or output may also be corrected for.

With a time-invariant plant the effective GPC control law is stationary when the receding horizon principle is applied (with a linear model and no hard constraints). This is in contrast with for instance the finite-horizon LQ controller in which the state feedback is time-varying. Hence, by using receding horizon control the time-invariance of stationary infinite horizon controllers is obtained but with a finite horizon criterion. See (Bitmead *et al.* 1990) for details and further comments.

4.2.2 Extensions to the basic GPC

A vast number of algorithms offering different extensions and modifications to the basic GPC scheme has been presented in the literature. Below we list the main themes of the research on predictive control and some references.

Because of the large number of publications on predictive control it would be inappropriate to give a complete survey here and the list is by no means complete.

Guaranteed stability

The basic GPC algorithm does not generally provide nominal stability. Some results for special cases have been given in (Clarke *et al.* 1987b, Clarke and Mohtadi 1989). A number of modifications have been suggested in order to get guaranteed nominal stability. The so-called Stable Generalized Predictive Control (SGPC) algorithm is presented in a deterministic setting in (Kouvaritakis *et al.* 1992). It works by applying GPC after closing a stabilizing feedback loop around the system to be controlled. A multivariable version, the MSGPC, is presented in (Kouvaritakis and Rossiter 1993). The Stabilizing I/O Receding Horizon Controller (SIORHC) presented in (Mosca and Zhang 1992, Bemporad *et al.* 1994) achieves stability results also in a deterministic setting by adding a zero terminal state constraint, that is the terminal value of the output must reach the desired setpoint. A similar idea is used in (Clarke and Scattolini 1991) in the so-called CRHPC (Constrained Receding Horizon Predictive Controller). In (Chisci *et al.* 1996) the approach is extended to a stochastic setting in a state-space framework. The resulting controller, the Stabilizing Stochastic Predictive Controller (SSPC) extends straightforwardly to MIMO plants. In (Demircioğlu and Clarke 1993) guaranteed stability is proved for the so-called GPCW where the GPC control criterion is augmented by an end-point state weighting.

Hard constraints

One of the advantages of the predictive control methodology is that hard constraint can be handled in a systematic way in the control optimization problem due to the use of the receding horizon principle. In (Tsang and

Clarke 1988) the GPC is extended with a quadratic programming (QP) approach to deal with rate constraints on the input. The QP approach is also discussed in (Soeterboek 1992, Camacho 1993). A constrained version of the SGPC, the CSGPC, is presented in (Rossiter and Kouvaritakis 1993) where hard constraints on the input rate and level and the output level are discussed. Feasibility and stability issues of the CSGPC are discussed in (Rossiter *et al.* 1994).

Adaptive predictive control

Model-based predictive control rely on the existence of an estimated system model by means of which output predictions can be computed. Hence, such algorithms are directly applicable in indirect adaptive control schemes where an estimation algorithm is combined with the control design block usually using the certainty equivalence principle, see (Åström and Wittenmark 1989). An analysis of the interaction between parameter estimation and control design in adaptive predictive controllers can be found in (Bitmead *et al.* 1990). Robustness results for an adaptive version of the SIORHC controller can be found in (Clarke *et al.* 1994). An adaptive version of the CRHPC algorithm with level and rate constraints on the input is analyzed in (De Nicolao *et al.* 1996).

Some advances in the development of a direct adaptive generalized predictive controller where the control law parameters are estimated directly have been reported in (Wang 1994, Wang and Henriksen 1994, Wang and Li 1995). The algorithm requires estimates of the impulse-response parameters of the plant. With some assumptions on the estimation error of these parameters global convergence results are obtained.

Another approach to adaptive predictive control is the MUSMAR algorithm, (Mosca *et al.* 1989, Mosca 1995, Chisci *et al.* 1994). The MUSMAR consists of a set of parallel RLS identifiers each providing a predictive model for the system output over the prediction horizon. Using such a multipredictor structure has the advantage of being less sensitive to unmodelled dynamics as the predictors are not computed indirectly from a single model.

Nonlinear predictive control

The predictive control concept can be applied equally well to nonlinear process models. However, the obtained optimization problems also become nonlinear and the computational approach will depend on the structure of the model. An algorithm based on a Hammerstein model is presented in (Zhu *et al.* 1991). A theoretical analysis of general problems can be found in (Henson and Seborg 1993).

Continuous-time predictive control

A continuous-time version of the GPC, the so-called CGPC, is presented in (Demircioğlu and Gawthrop 1991). Prediction of the output is obtained by a continuous-time emulator-based predictor which was discussed in Section 3.4. The CGPC is algebraically very similar to the discrete-time GPC algorithm. However, the interpretation of the design variables is different. The CGPC is discussed in more detail in the next section. A multivariable version of the CGPC is presented in (Demircioğlu and Clarke 1992). Two modified versions of CGPC, the CGPC_C and CGPC_W, are presented in (Demircioğlu and Clarke 1992). The CGPC_C includes an end-point state constraint similar to the discrete-time CRHPC algorithm. The CGPC_W includes an end-point state-weight in the criterion function similar to the GPCW algorithm. Both the CGPC_C and CGPC_W have guaranteed stability properties.

4.3 Delta-domain emulator-based GPC

In this section we develop long range predictive control algorithms formulated in the δ-domain. As the δ-operator is simply a linear transformation of the shift-operator it is of course possible to find an exact δ-domain transformation of the q-domain GPC, see (Thygesen 1993). Here we use the emulator-based predictor presented in Section 3.4 to emphasize the close connection between discrete-time and continuous-time theory. Using this predictor we develop a general unified framework for discrete-time and continuous-time predictive control algorithms.

Although the control algorithms presented here are based on discrete-time δ-domain ARMAX models, the continuous nature of the emulator-based predictor allows us to consider continuous-time like integral-based control criteria of the type

$$J = \mathcal{E}\left\{ \int_{T_1}^{T_2} (y(t) - w(t))^2 \; dt + \rho \int_0^{T_2 - T_1} u^2(t) \, dt \right\} \tag{4.8}$$

In the presented algorithm we circumvent determination of the conditional expectation by replacing $y(t)$ with the output predicted by the emulator-based predictor.

The connection between emulator-based prediction and optimal prediction established in Section 3.4.5 makes it possible to derive exact equivalents of discrete-time control algorithms based on the usual summation-based criteria of the type

$$J = \mathcal{E}\left\{ \sum_{i=N_1}^{N_2} [y(t+iT) - w(t, iT)]^2 + \rho \sum_{i=0}^{N_2 - N_1} [u(t, iT)]^2 \right\} \tag{4.9}$$

However, the possibility of choosing the predictor N_y of the emulator-based predictor lower than the actual prediction horizon N_2 gives a potential computational advantage at the cost of an approximation error in the criterion function. This computational advantage may be significant in adaptive control where computational speed is of importance.

The above criteria are closely related to the continuous-time and discrete-time optimal linear quadratic tracking controllers. However, we consider implementation using the receding horizon principle as described in Section 4.2.1. Also we treat the integral-based and the summation-based controller in a unified manner.

A number of extensions to these two basic type of control algorithms are also presented. We equip the control criteria with frequency weighting on the input and output signals. This gives very flexible control algorithms. It is also shown how an exact mathematical equivalent of the usual q-domain GPC's may be achieved by the introduction of a new general constraint on future control input.

A state-space formulation of the basic control algorithms is presented. This formulation reveals how the controllers may be interpreted as a state-feedback combined with a state-estimator (observer), which is convenient for implementation. The use of state-space models gives a natural approach to multivariable extensions of the presented algorithms. Two possible formulations of such extensions are presented.

We start by introducing the two new control criteria that will be examined. As for the emulator-based predictor the presented control algorithms will be derived for δ-domain ARMAX models of the form

$$A(\delta)y(t) = B(\delta)u(t) + C(\delta)e(t) \tag{4.10}$$

where $\deg(A) = n$, $\deg(B) = m < n$ and $\deg(C) = n - \Delta$ where $\Delta \in \{0, 1\}$. Further information on this choice of model can be found in Section 3.2 and Section 3.3. Other more general model structures, for instance the ARIMAX model can be rewritten to ARMAX form.

4.3.1 Delta-domain approximation of CGPC

Consider the following control criterion

$$J_C(T_1, T_2, N_y, \rho) = \int_{T_1}^{T_2} [\hat{y}^*(t+\tau) - w(t,\tau)]^2 \, d\tau + \rho \int_0^{T_2 - T_1} [u(t,\tau)]^2 \, d\tau \tag{4.11}$$

where $\hat{y}^*(t+\tau)$ is the emulator-based prediction of $y(t+\tau)$. The future reference $w(t,\tau)$ and input $u(t,\tau)$ are approximated using expansions similar to (3.93).

The resulting controller minimizing this criterion will be referred to as the DCGPC (Delta-domain approximation of CGPC). This algorithm was first presented in (Lauritsen *et al.* 1994a, Lauritsen *et al.* 1995a). As the name indicates the DCGPC control criterion is very closely related to that of the continuous-time GPC presented in (Demircioğlu and Gawthrop 1991). As remarked in Chapter 3 the emulator-based predictor converges, as the sampling period tends to zero, to the continuous-time so-called T-ahead predictor used in (Demircioğlu and Gawthrop 1991). As a result the continuous-time limit of the criterion J_C corresponds to the criterion used in the CGPC apart from the way that reference-tracking is obtained. The CGPC generates the future reference trajectory as the output of a reference model so that the reference trajectory is a smooth transition from the output $y(t)$ to the setpoint. As analyzed in (Soeterboek 1992) using such a procedure affects the regulator behaviour and robustness and may in some situations cause instability. An alternative is to introduce frequency weighting of the system output in the control criterion. This is discussed in Section 4.4.

One of the advantages of formulating this discrete-time version of the continuous-time CGPC controller, is the fact that the resulting optimal controller can be implemented without approximation, cf. the discussion in Section 2.5.3.

Usually in discrete-time control an integral-based criterion like the above is not feasible as the discrete-time models only deliver signal values at the sampling instants. However, the continuous nature of the emulator-based predictor presented in Section 3.4 makes it possible to approximate signal values in between sampling instants. With fast sampling, that is with a sampling frequency chosen high enough to cover all relevant frequency contents in the closed-loop signals, the continuous approximation/prediction of the signals from a discrete-time model can be done without risking problems of unpredictable intersample behaviour.

An interesting observation is that the above control criterion is in fact independent of the sampling period. There is an implicit dependence through the emulator-based predictor as the approximation errors in the continuous prediction of the output depend on the sampling period. However, as will be illustrated, the DCGPC controller is quite insensitive to the choice of sampling period.

4.3.2 Delta-domain approximation of q-domain GPC

An approximation of the usual q-domain GPC is obtained by the following control criterion

$$J_Q(N_1, N_2, N_y, \rho) = \sum_{i=N_1}^{N_2} [\hat{y}^*(t+iT) - w(t,iT)]^2 + \rho \sum_{i=0}^{N_2-N_1} [u(t,iT)]^2 \tag{4.12}$$

The resulting controller will be referred to as the DQGPC controller. The DQGPC controller was first presented in (Lauritsen *et al.* 1994a, Lauritsen *et al.* 1995a).

This criterion has a strong resemblance to a positional version of the usual q-domain GPC where the control signal itself and not the increment is punished by the criterion function. Compare with Section 4.2.1. We will later equip the criterion with frequency filtering to allow for instance weighting of control increments.

A similar control algorithm has been presented in (Boucher *et al.* 1993, Dumur and Boucher 1994, Bleile *et al.* 1995). They consider the following transformation of output δ-derivatives to output predictions

$$y(t, kT) = \sum_{i=0}^{N_y} \binom{k}{i} T^i \delta^i y(t) \tag{4.13}$$

For $N_y = k$ this is the exact transformation considered in Theorem 11. Unlike the transformation used in the δ-domain emulator-based predictor this transformation is only applicable at sampling instants.

4.3.3 The optimal control law

Emulator-based output prediction

The future system output is predicted by means of the emulator-based predictor presented in Chapter 3. According to (3.156) we can then write

$$\hat{y}^*(t+\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\left(\tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}} + \boldsymbol{y}^0\right) \tag{4.14}$$

where $\boldsymbol{T}_{N_y}(\tau)$ is a vector depending only on the predictor order N_y and τ, $\boldsymbol{M}_{s\gamma}(N_y)$ is a conversion matrix depending only on N_y, $\tilde{\boldsymbol{G}}$ depends on the model parameters through two coupled Diophantine equations, $\tilde{\boldsymbol{u}}$ is a vector of future control input δ-derivatives and $\boldsymbol{y}^0$ is a signal depending on the model parameters and past input/output signal values. As discussed in Section 3.2.2 the use of incomplete or complete information is conditioned on the relative degree of the noise model, that is Δ.

Control constraint

Both criteria J_C and J_Q are subject to the constraint

$$\delta^k u(t) = 0 \quad \text{for } k > N_u \tag{4.15}$$

which constrains the smoothness of the control strategy. The *control order* N_u satisfies

$$0 \leq N_u \leq N_y - d \tag{4.16}$$

By defining

$$\boldsymbol{u} = \left[u(t)\ \delta u(t)\ \delta^2 u(t)\ \ldots\ \delta^{N_u} u(t)\right]^\top \tag{4.17}$$

we can, subject to the constraint, write the emulator based predictor as

$$\hat{y}^*(t+\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\left(\boldsymbol{G}\boldsymbol{u} + \boldsymbol{y}^0\right) \tag{4.18}$$

where $\boldsymbol{G}$ is obtained as the first $N_u + 1$ columns of $\tilde{\boldsymbol{G}}$.

The constraint has the effect of reducing the dimension of the input vector which is the optimization variable in the control design. This can be utilized to ensure solvability in cases where the full problem ($N_u = \tilde{N}_u$) is not solvable. The control order N_u can also be regarded as an important design variable which can be used to tune the closed-loop responses. Roughly speaking, the more difficult the system (non-minimumphase, instability, etc.) the higher N_u is required to give satisfactory closed-loop responses.

Unified control criterion

We express the future reference and control signals as Taylor-series expansions similar to (3.93)

$$w(t,\tau) \triangleq \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{w} \tag{4.19}$$

$$u(t,\tau) \triangleq \boldsymbol{T}_{N_u}(\tau)\boldsymbol{M}_{s\gamma}(N_u)\boldsymbol{u} \tag{4.20}$$

where

$$\boldsymbol{w} = \left[w(t)\ \delta w(t)\ \delta^2 w(t)\ \ldots\ \delta^{N_y} w(t)\right]^\top \tag{4.21}$$

Note, that according to Theorem 10 these expansions are actually exact at the sampling instants, that is $w(t, kT) = w(kT)$ with k being an integer less than or equal to N_y. With these definitions and using (4.18) the DCGPC control criterion can be expressed as

$$\begin{aligned} J_C &= \int_{T_1}^{T_2} \left[\boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\left(\boldsymbol{Gu}+\boldsymbol{y}^0-\boldsymbol{w}\right)\right]^2 d\tau \\ &\qquad + \rho \int_0^{T_2-T_1} \left[\boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{u}\right]^2 d\tau \end{aligned} \tag{4.22}$$

and the DQGPC criterion as

$$\begin{aligned} J_Q &= \sum_{i=N_1}^{N_2} \left[\boldsymbol{T}_{N_y}(iT)\boldsymbol{M}_{s\gamma}(N_y)\left(\boldsymbol{Gu}+\boldsymbol{y}^0-\boldsymbol{w}\right)\right]^2 \\ &\qquad + \rho \sum_{i=0}^{N_2-N_1} \left[\boldsymbol{T}_{N_y}(iT)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{u}\right]^2 \end{aligned} \tag{4.23}$$

By moving quantities which do not depend on the integration or summation variable outside the integration or summation it is not difficult to see, that a unified criterion can be specified as

$$J_U = \left(\boldsymbol{Gu}+\boldsymbol{y}^0-\boldsymbol{w}\right)^\top \boldsymbol{W}_{\mathrm{y}} \left(\boldsymbol{Gu}+\boldsymbol{y}^0-\boldsymbol{w}\right) + \rho\boldsymbol{u}^\top\boldsymbol{W}_{\mathrm{u}}\boldsymbol{u} \tag{4.24}$$

where the weighting matrices are given by

$$\boldsymbol{W}_{\mathrm{y}} = \begin{cases} \boldsymbol{M}_{s\gamma}^\top(N_y) \displaystyle\int_{T_1}^{T_2} \boldsymbol{T}_{N_y}^\top(\tau)\boldsymbol{T}_{N_y}(\tau)\, d\tau\ \boldsymbol{M}_{s\gamma}(N_y) \\ \boldsymbol{M}_{s\gamma}^\top(N_y) \displaystyle\sum_{i=N_1}^{N_2} \boldsymbol{T}_{N_y}^\top(iT)\boldsymbol{T}_{N_y}(iT)\boldsymbol{M}_{s\gamma}(N_y) \end{cases} \tag{4.25}$$

$$\boldsymbol{W}_{\mathrm{u}} = \begin{cases} \boldsymbol{M}_{s\gamma}^\top(N_u) \displaystyle\int_{0}^{T_2-T_1} \boldsymbol{T}_{N_u}^\top(\tau)\boldsymbol{T}_{N_u}(\tau)\, d\tau\ \boldsymbol{M}_{s\gamma}(N_u) \\ \boldsymbol{M}_{s\gamma}^\top(N_u) \displaystyle\sum_{i=0}^{N_2-N_1} \boldsymbol{T}_{N_u}^\top(iT)\boldsymbol{T}_{N_u}(iT)\boldsymbol{M}_{s\gamma}(N_u) \end{cases} \tag{4.26}$$

with uppermost entry belonging to J_C in both cases. Notice the close analogy between the DCGPC case and the DQGPC case. It can be shown, that

$\boldsymbol{W}_{\mathrm{u}}$ is positive definite for $T_2 - T_1 > 0$ in the DCGPC case and $T > 0$ and $N_2 - N_1 \geq N_u$ and the DQGPC case, see (Lauritsen *et al.* 1994a).

The gradient of J_U with respect to the control vector $\boldsymbol{u}$ is

$$\frac{\partial J_U}{\partial \boldsymbol{u}} = 2\boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}} \boldsymbol{G}\boldsymbol{u} + 2\boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}} (\boldsymbol{y}^0 - \boldsymbol{w}) + 2\rho \boldsymbol{W}_{\mathrm{u}} \boldsymbol{u} \tag{4.27}$$

and the Hessian

$$\frac{\partial^2 J_U}{\partial \boldsymbol{u}^2} = 2\boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}} \boldsymbol{G} + 2\rho \boldsymbol{W}_{\mathrm{u}} \tag{4.28}$$

Note, that if $\rho = 0$ then the definiteness properties of the Hessian depend on the system through $\boldsymbol{G}$. In this situation the Hessian can not be guaranteed to be positive definite. On the other hand if $\rho > 0$ then the Hessian is always positive definite and hence setting the gradient equal to zero and solving for $\boldsymbol{u}$ always gives the global optimum of the minimization problem. One obtains the following optimal future control vector.

$$\begin{aligned} \boldsymbol{u} &= \left(\boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}} \boldsymbol{G} + \rho \boldsymbol{W}_{\mathrm{u}}\right)^{-1} \boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}} \left(\boldsymbol{w} - \boldsymbol{y}^0\right) \\ &= \boldsymbol{K} \left(\boldsymbol{w} - \boldsymbol{y}^0\right) \end{aligned} \tag{4.29}$$

where

$$\boldsymbol{K} = \left(\boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}} \boldsymbol{G} + \rho \boldsymbol{W}_{\mathrm{u}}\right)^{-1} \boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}} \tag{4.30}$$

Note, that in the case $N_u = 0$ the matrix to be inverted is a scalar, which is highly attractive from a computational point of view in adaptive applications. The controller is implemented following a receding horizon strategy and hence only the first element of $\boldsymbol{u}$ needs to be calculated:

$$u(t) = \boldsymbol{k}^\top \left(\boldsymbol{w} - \boldsymbol{y}^0\right) \tag{4.31}$$

where $\boldsymbol{k}^\top$ is the first row of $\boldsymbol{K}$ with coefficients

$$\boldsymbol{k}^\top = \left[k_0 \ k_1 \ \ldots \ k_{N_y}\right] \tag{4.32}$$

4.3.4 External control law

The controllers can easily be rewritten to the polynomial form

$$u(t) = \frac{Q(\delta)}{R(\delta)}w(t) - \frac{S(\delta)}{R(\delta)}y(t) \tag{4.33}$$

Take (4.31) and insert the vector definitions (4.32), (4.21), (3.150) and (3.147). Then

$$u(t) = \sum_{i=0}^{N_y} k_i \left(\delta^i w(t) - H_e^{[i]}(\delta)\frac{u(t)}{C(\delta)} - F_e^{[i]}(\delta)\frac{y(t)}{C(\delta)} \right) \tag{4.34}$$

This leads to

$$\begin{aligned} &\left(C(\delta) + \sum_{i=0}^{N_y} k_i H_e^{[i]}(\delta) \right) u(t) = \\ &\qquad C(\delta)\left(\sum_{i=0}^{N_y} k_i \delta^i \right) w(t) - \left(\sum_{i=0}^{N_y} k_i F_e^{[i]}(\delta) \right) y(t) \end{aligned} \tag{4.35}$$

This is seen to be on the form (4.33) with

$$\begin{aligned} R(\delta) &= C(\delta) + \sum_{i=0}^{N_y} k_i H_e^{[i]}(\delta) \\ Q(\delta) &= C(\delta)K(\delta) \\ S(\delta) &= \sum_{i=0}^{N_y} k_i F_e^{[i]}(\delta) \end{aligned} \tag{4.36}$$

where

$$K(\delta) = k_0 + k_1\delta + \cdots + k_{N_y}\delta^{N_y} \tag{4.37}$$

and

$$\deg(R) = n - \Delta \tag{4.38}$$
$$\deg(Q) = n + N_y - \Delta \tag{4.39}$$
$$\deg(S) = n - 1 \tag{4.40}$$

Note, that the transfer function $S(\delta)/R(\delta)$ is strictly proper if $\Delta = 0$ which means, that the last value of the output $y(t)$ is not fed back, i.e. incomplete information is used. This is modelling case $\mathcal{M}_2$, see Section 3.2.2. If on the other hand $\Delta = 1$, then $S(\delta)/R(\delta)$ is exactly proper and complete information is used. This is modelling case $\mathcal{M}_1$. Note also, that the transfer function $Q(\delta)/R(\delta)$ is non-causal, i.e. knowledge of future δ-derivatives of the reference signal $w(t)$ is necessary. As discussed in Section 4.2.1, this information may be accessible in some applications, e.g. robotics. In other applications it may be necessary to make some assumptions on the future reference. The common constant future reference assumption can be obtained by choosing

$$\boldsymbol{w} = [w(t)\ 0\ \ldots\ 0]^\top \tag{4.41}$$

In this case the $Q(\delta)$ polynomial becomes

$$Q(\delta) = k_0 C(\delta) \tag{4.42}$$

with $\deg(Q) = n - \Delta$ so that the transfer function $Q(\delta)/R(\delta)$ becomes exactly proper.

4.3.5 Closed-loop expressions

The closed-loop system is illustrated in the block diagram Figure 4.1. From the figure it is easy to derive the following closed-loop expressions

$$\begin{aligned} y(t) &= \frac{B(\delta)Q(\delta)}{A(\delta)R(\delta) + B(\delta)S(\delta)} w(t) + \frac{R(\delta)C(\delta)}{A(\delta)R(\delta) + B(\delta)S(\delta)} e(t) \\ &= \frac{B(\delta)Q(\delta)}{C_{cl}(\delta)} w(t) + \frac{R(\delta)C(\delta)}{C_{cl}(\delta)} e(t) \end{aligned} \tag{4.43}$$

where the closed-loop characteristic polynomial is given as

$$C_{cl}(\delta) = A(\delta)R(\delta) + B(\delta)S(\delta) \tag{4.44}$$

The noise polynomial $C(\delta)$ can be interpreted as an observer polynomial since it cancels in the transfer function of the deterministic part of the

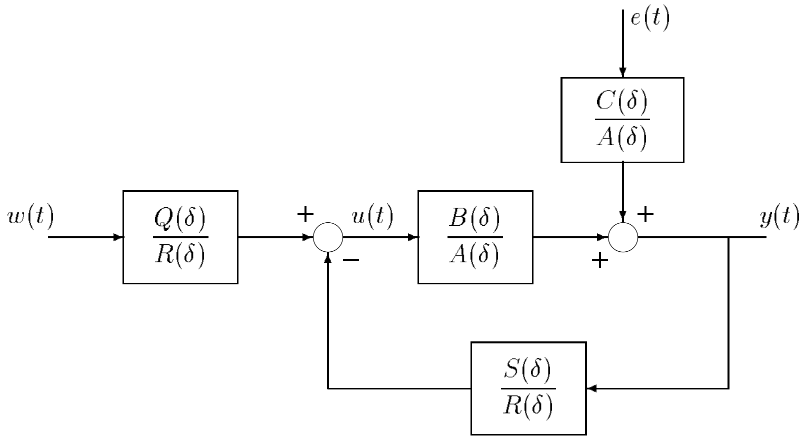

Figure 4.1: Total closed loop system with two degrees of freedom external δ-domain controller.

system. This becomes clear from the following rewritings

$$\begin{aligned} C_{cl}(\delta) &= A(\delta)\left(C(\delta)+\sum_{i=0}^{N_y} k_i H_e^{[i]}(\delta)\right)+B(\delta)\sum_{i=0}^{N_y} k_i F_e^{[i]}(\delta) \\ &= C(\delta)A(\delta)+\sum_{i=0}^{N_y} k_i\left(A(\delta)H_e^{[i]}(\delta)+B(\delta)F_e^{[i]}(\delta)\right) \qquad (4.45) \end{aligned}$$

From (3.127) and (3.134) we get

$$\begin{aligned} &A(\delta)H_e^{[i]}(\delta)+B(\delta)F_e^{[i]}(\delta) \\ &\quad = A(\delta)\left(E_e^{[i]}(\delta)B(\delta)-C(\delta)G_e^{[i]}(\delta)\right)+B(\delta)\left(\delta^i C(\delta)-A(\delta)E_e^{[i]}(\delta)\right) \\ &\quad = C(\delta)\left(\delta^i B(\delta)-A(\delta)G_e^{[i]}(\delta)\right) \qquad (4.46) \end{aligned}$$

Using this gives

$$C_{cl}(\delta) = C(\delta)\left(A(\delta)+\sum_{i=0}^{N_y} k_i\left(\delta^i B(\delta)-A(\delta)G_e^{[i]}(\delta)\right)\right)$$

	Description	Restrictions
T_1	Minimum prediction horizon	$T_1 \geq 0$
T_2	Maximum prediction horizon	$T_2 > T_1$
N_1	Minimum costing horizon	$N_1 \geq 1$
N_2	Maximum costing horizon	$N_2 \geq N_1$
N_y	Output predictor order	$N_y \geq d$
N_u	Control order	$0 \leq N_u \leq N_y - d$
ρ	Control weighting factor	$\rho \geq 0$
Δ	Relative degree of noise model	$\Delta \in \{0, 1\}$
$C(\delta)$	Noise model	$\deg(C) = n - \Delta$
T	Sampling period	

Table 4.1: DCGPC and DQGPC design parameters.

$$= \quad C(\delta)A_m(\delta) \tag{4.47}$$

Inserting this into (4.43) yields

$$y(t) = \frac{B(\delta)\sum_{i=0}^{N_y} k_i\delta^i}{A_m(\delta)}w(t) + \frac{C(\delta) + \sum_{i=0}^{N_y} k_i H_e^{[i]}(\delta)}{A_m(\delta)}e(t) \tag{4.48}$$

Hence, $C(\delta)$ is cancelled in the transfer function from $w(t)$ to $y(t)$ and thus does not affect the deterministic part of the system. It does affect the noise part of the system as it enters into the transfer function from $e(t)$ to $y(t)$.

4.3.6 Design Parameters

A list of design parameters for the DCGPC and DQGPC algorithms is shown in Table 4.1. The choice of each of these parameters is now briefly discussed.

T_1 is a DCGPC design parameter and can always be chosen to be zero. If the system is known to have a time-delay then T_1 can be set equal to the delay. If the delay is not known exactly then T_1 can be set equal to the maximum time-delay, see also the discussion in (Demircioğlu and Gawthrop 1991) for the CGPC. For non-minimum phase systems T_1 may be chosen larger than zero to exclude the negative going part of the output caused by step inputs.

T_2 is a DCGPC design parameter and can be used to tune the output response. Small T_2 gives fast output responses and large values cause slower responses.

N_1 is a DQGPC design parameter and will affect the controller quite similarly to T_1. It is similar to the minimum costing horizon in the usual GPC, see (Clarke *et al.* 1987a).

N_2 is a DQGPC design parameter with a similar effect on the closed-loop as T_2. As for N_1 discussions on this maximum costing horizon from the usual GPC applies to the DQGPC as well, see (Clarke *et al.* 1987a).

N_y is strongly related to T_2 or N_2. The higher the maximum prediction horizon the higher predictor order is necessary. Including a large number of terms in the Taylor-series expansions will make it possible to describe variations further into the future. Also complex variations of the system output will call for a high value of N_y. For instance non-minimum phase system which are known have a negative going part in the step response require higher values of N_y. Typical values of N_y are 6–30.

N_u can be used as a tuning knob to adjust the output response. A high value of N_u eases the constraint on the projected future control input which allows more variation in the control input. Hence a large value of N_u gives more active control than a small value, at the cost of computational effort. Complex system may require an increased value of N_u. Typically N_u is 0–3.

ρ may be used to reduce the power consumption of the control signal if necessary, for instance to keep the input signal within operation limits. This will naturally give less active control. However, because the control signal itself and not increments (as e.g. in the GPC) is weighted in the control criterion, a positive ρ will lead to steady-state errors for processes with no "built-in" integration (type 0 processes). This can be avoided by introducing suitable filtering of the input signal in the control criterion, see Section 4.4. Often ρ is chosen to be zero because it is difficult to tune. Note, that control weighting is not always necessary for the control of non-minimum phase systems as in other types of control algorithms to avoid cancelling of unstable zeros. A positive value of ρ can also be introduced to ensure a positive definite Hessian in the criterion minimization and thus to ensure solvability of the control design problem.

$C(\delta)$ has implications on the disturbance rejection and robustness of the closed-loop system, similar to the q-domain GPC. Often the noise polynomial is considered a design variable—the observer polynomial.

T Choice of sampling period is of great importance in any sampled control system. The interesting point in this context is how the controller behaves if the sampling rate is not chosen unreasonable fast. It was illustrated in Section 3.5 that the emulator-based predictor behaves well even with relatively slow sampling. This makes the choice of T less critical as compared to for instance control algorithms formulated in the q-domain where fast sampling is a problem. Here we can sample very fast or quite slow without problems.

The design variables above a very similar to those used in the continuous-time GPC and the basic GPC for the DCGPC case and DQGPC case respectively. For simulation examples on the effect of the design variables and guidelines on how to choose these variables we refer to (Demircioğlu and Gawthrop 1991) and (Soeterboek 1992). We will however illustrate the advantages of the proposed algorithms in the following simulation examples.

T	DCGPC-poles (s-domain)
0.0001	$-1.699,\ -1.113 \pm 1.492\,i$
0.001	$-1.699,\ -1.114 \pm 1.492\,i$
0.01	$-1.703,\ -1.115 \pm 1.495\,i$
0.05	$-1.720,\ -1.122 \pm 1.509\,i$
0.1	$-1.743,\ -1.130 \pm 1.528\,i$
0.2	$-1.793,\ -1.148 \pm 1.569\,i$
0.3	$-1.853,\ -1.167 \pm 1.617\,i$
0.4	$-1.923,\ -1.186 \pm 1.673\,i$
0.5	$-2.010,\ -1.205 \pm 1.740\,i$

Table 4.2: Closed-loop DCGPC poles (s-domain) for various sampling periods.

4.3.7 Simulation examples

The DCGPC controller

Example 6 *To show that the sampling period has very little influence on the closed-loop DCGPC poles consider the third order non-minimum phase system*

$$\frac{B(s)}{A(s)} = \frac{-0.2s+1}{s(s^2+1)} \tag{4.49}$$

controlled by the DCGPC controller with design parameters

$$\begin{array}{l} T_1 = 0,\ T_2 = 5,\ T = 0.0001, 0.001, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 \\ N_y = 30,\ N_u = 3,\ \rho = 0 \end{array} \tag{4.50}$$

The predictor order N_y has been chosen quite high so as not to affect the pole locations. The pole-locations are shown in Table 4.2 and plotted in Figure 4.2. The poles have been converted to the s-domain using (3.82) since the δ-domain stability region depends on T which prevents a direct comparison of the δ-domain poles.

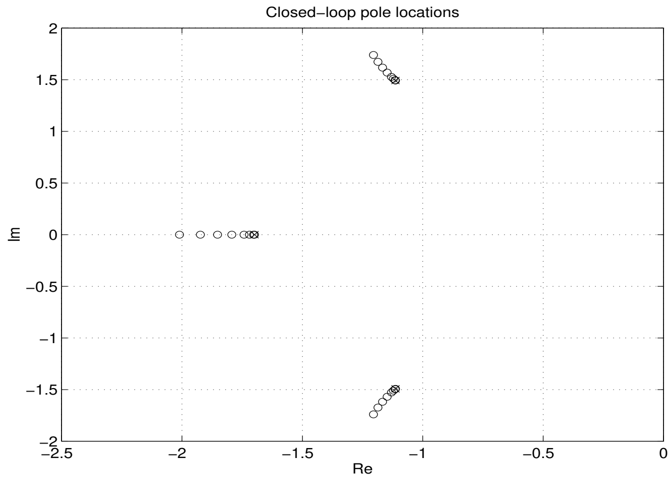

Figure 4.2: Closed-loop DCGPC poles (s-domain) corresponding to Table 4.2.

The figure shows, that the poles vary very little with the sampling period. Hence, even with quite large sampling periods the DCGPC controller is very close to the continuous-time limit obtained when $T \to 0$. *Note that different rules of thumb (Åström and Wittenmark 1990) suggest a sampling period of around 0.1, so* $T = 0.5$ *is very slow sampling.*

In Figure 4.3 the discrete-time closed-loop step-responses are shown for $T = 0.01$, $T = 0.1$ *and* $T = 0.2$. *We see a very close accordance between the three step-responses. Responses for faster sampling rates are not distinguishable from* $T = 0.01$. □

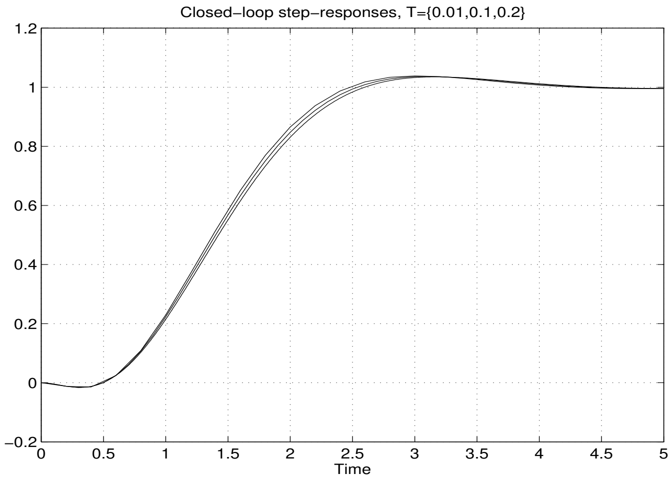

Figure 4.3: Closed-loop responses for varying T.

The DQGPC controller

Example 7 *Consider the basic GPC criterion*

$$J_{GPC} = E\left\{\sum_{i=N_1}^{N_2} [y(t+iT) - w(t+iT)]^2\right\} \tag{4.51}$$

$$= const + \sum_{i=N_1}^{N_2} [\hat{y}(t+iT) - w(t+iT)]^2 \tag{4.52}$$

where control weighting has been omitted and $\hat{y}(t+iT)$ is the minimum variance prediction of $y(t+iT)$ given information up to time t. The criterion is subject to the constant future control constraint

$$u(t+iT) = u(t+N_uT), \quad i > N_u \tag{4.53}$$

When comparing the above version of the GPC based on an ARMAX model with the DQGPC one observes a strong resemblance. A difference is, that

T	N_2	Exact GPC	DQGPC
0.01	100	$-3.4572,\ -1.6502 \pm 2.1329\,i$	$-3.4595,\ -1.6506 \pm 2.1325\,i$
0.05	20	$-3.4707,\ -1.6507 \pm 2.1498\,i$	$-3.4706,\ -1.6507 \pm 2.1498\,i$
0.1	10	$-3.4920,\ -1.6526 \pm 2.1768\,i$	$-3.4920,\ -1.6526 \pm 2.1768\,i$
0.2	5	$-3.5529,\ -1.6590 \pm 2.2516\,i$	$-3.5529,\ -1.6590 \pm 2.2516\,i$
0.5	2	$-3.9929,\ -1.6759 \pm 2.7130\,i$	$-3.9929,\ -1.6759 \pm 2.7130\,i$

Table 4.3: Closed-loop DQGPC and GPC poles (s-domain) for various sampling periods.

when $N_u > 0$ the future control constraints in (4.53) and (4.15) are not compatible. However, for $N_u = 0$ the constraints are equal and the DQGPC approximates the above version of the GPC. To illustrate this some control designs have been made with the system (3.182)

$$\frac{B(s)}{A(s)} = \frac{3}{(s+3)(s^2+s+1)} \tag{4.54}$$

and design parameters

$$\begin{aligned} &N_1 = 1,\ N_2T = 10,\ T = 0.01, 0.05, 0.1, 0.2, 0.5 \\ &N_y = 10,\ N_u = 0,\ \rho = 0 \end{aligned} \tag{4.55}$$

Note, that the maximum absolute prediction horizon N_2T is kept constant so that N_2 varies with T. The closed-loop poles (transformed to the s-domain) for both the exact GPC and the approximating DQGPC are shown in Table 4.3.

The table shows, that when $N_y \geq N_2$ the two controllers are identical with the given precision. When $N_2 > N_y$ a very small discrepancy between the exact GPC and the DQGPC appears. However, the DQGPC is less computationally demanding since, for instance, at $T = 0.01$ only 10 δ-derivatives need to be emulated in the DQGPC whereas the exact GPC needs to predict 100 samples ahead. The extrapolating ability of the new δ-domain emulator-based predictor clearly gives a potential computational advantage when N_2 is large. □

4.4 Frequency weighting

An important extension to the basic DCGPC and DQGPC schemes is the inclusion of frequency weighting in the control criteria. In this section the basic criteria will be equipped with separate linear filters on the system output and the control input. Such filters allow a specific frequency dependent tuning of the controller.

The output filter can be used to tune the output response. With certain choices of design parameters this filter can be considered an approximate inverse model of the desired output response.

The input filter may be used to avoid excitation of the system in certain frequency areas. For instance, undesired excitation at resonance frequencies may be attenuated by selecting an inverse notch filter as the frequency weight on the system input. Also the influence of high-frequency measurement noise may be reduced by proper frequency weighting. Some examples for long range predictive control algorithms are given in e.g. (Soeterboek 1992). Another important use of the input filter is the possibility to weight control increments as in the GPC algorithm described in Section 4.2.1. In this way steady-state errors due to control weighting can be avoided for processes with no integration.

Another possible use of filters in the design criterion is illustrated in the description of the Zang-scheme, see Section 5.7.3. There the filters are data-induced and used as an integral part of an iterative control scheme.

In addition to the extended frequency weighted control criterion a more general constraint on future control input is proposed. As will be shown this allows us to incorporate the usual constant future control known from q-domain GPC's in the δ-domain emulator-based GPC formulation. As a result we can get a mathematically equivalent formulation of the usual GPC algorithm in our δ-domain GPC framework.

4.4.1 Criterion functions

Consider the DCGPC or the DQGPC control criterion with filtered input and output signals

$$J_C = \int_{T_1}^{T_2} \left[\hat{\zeta}^*(t+\tau) - w(t,\tau)\right]^2 d\tau + \rho \int_0^{T_2-T_1} [u_f(t,\tau)]^2 \, d\tau \tag{4.56}$$

and

$$J_Q = \sum_{i=N_1}^{N_2} \left[\hat{\zeta}^*(t+iT) - w(t,iT)\right]^2 + \rho \sum_{i=0}^{N_2-N_1} [u_f(t,iT)]^2 \tag{4.57}$$

where $\zeta^*(t+\tau)$ is an emulator-based prediction the filtered version of the output

$$\zeta(t) = \frac{P_n(\delta)}{P_d(\delta)} y(t) \tag{4.58}$$

and where $u_f(t,iT)$ is an approximation of the filtered input

$$u_f(t) = \frac{Q_n(\delta)}{Q_d(\delta)} u(t) \tag{4.59}$$

Furthermore,

$$\begin{aligned} w(t,\tau) &= \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{w} &(4.60)\\ u_f(t,\tau) &= \boldsymbol{T}_{\tilde{N}_u}(\tau)\boldsymbol{M}_{s\gamma}(\tilde{N}_u)\boldsymbol{u}\mathrm{f} &(4.61)\end{aligned}$$

with

$$\boldsymbol{u}\mathrm{f} = \left[u_f(t) \;\; \delta u_f(t) \;\; \ldots \;\; \delta^{\tilde{N}_u} u_f(t)\right]^\top \tag{4.62}$$

The integer $\tilde{N}_u$ is the number of input δ-derivatives that have influence on the solution of the minimization problem and is defined later. In the basic unfiltered versions of these controllers as presented in Section 4.3, the approximation of future control input was based on only the first N_u δ-derivatives of $u(t)$. This made sense because the constraint

$$\delta^k u(t) = 0, \quad k > N_u \tag{4.63}$$

simply nullifies δ-derivatives of order higher than N_u. However, in this section we consider filtered versions of the control input and a more general constraint on future input δ-derivatives which will be introduced in Section 4.4.3. Hence, δ-derivatives of order higher than N_u of the filtered control input can no longer be disregarded.

There is no need to have a filter on the reference signal as such filtering can be achieved simply by feeding a filtered reference into the controller.

4.4.2 The filtered output

The filtered output is given by

$$\zeta(t) = \frac{P_n(\delta)}{P_d(\delta)} y(t) \tag{4.64}$$

where $\deg(P_d) = n_p$, $\deg(P_n) = n_p + r_p$ and the relative degree of P_d/P_n is $r_p \geq 0$, i.e. $\deg(P_n) \geq \deg(P_d)$. The filtered output can be expressed in terms of the control vector $\boldsymbol{u}$ and old input and output in a way similar to that used for the unfiltered version of the output, see (Lauritsen *et al.* 1995b). Hence, by means of the two coupled Diophantine-equations

$$\frac{\delta^k P_n(\delta)C(\delta)}{P_d(\delta)A(\delta)} = E_e^{[k]}(\delta) + \frac{F_e^{[k]}(\delta)}{P_d(\delta)A(\delta)} \tag{4.65}$$

$$\frac{E_e^{[k]}(\delta)B(\delta)}{C(\delta)} = G_e^{[k]}(\delta) + \frac{H_e^{[k]}(\delta)}{C(\delta)} \tag{4.66}$$

with

$$\deg(E_e^{[k]}) = \max\{k + r_p - \Delta, 0\} \tag{4.67}$$

$$\deg(F_e^{[k]}) = n + n_p - 1 \tag{4.68}$$

$$\deg(G_e^{[k]}) = \max\{k + r_p - d, 0\} \tag{4.69}$$

$$\deg(H_e^{[k]}) = \min\{n - 1 - \Delta, m + \deg(E_e^{[k]})\} \tag{4.70}$$

we can write the filtered k'th order output δ-derivative as

$$\zeta_k(t) = \underbrace{G_e^{[k]}u(t)}_{\text{future}} + \underbrace{\frac{H_e^{[k]}}{C}u(t) + \frac{F_e^{[k]}}{P_d C}y(t)}_{\text{past}} + \underbrace{E_e^{[k]}e(t)}_{\text{future}} \tag{4.71}$$

The emulation of $\zeta_k(t)$ is now obtained by taking expectation conditioned on complete or incomplete information depending on Δ, see Section 3.2.2

$$\begin{aligned} \zeta_k^*(t) &= \mathcal{E}\{\zeta_k(t)|\mathcal{I}_{\Delta,t,k}\} \\ &= G_e^{[k]}u(t) + \frac{H_e^{[k]}}{C}u(t) + \frac{F_e^{[k]}}{P_d C}y(t) \\ &= G_e^{[k]}u(t) + \zeta_k^0(t) \end{aligned} \tag{4.72}$$

Included in the information set $\mathcal{I}_{\Delta,t,k}$ is all relevant future control input signals. These are of course not available but are the optimization variables in the control design and are as such free to choose.

The so-called *free emulation* is given as

$$\zeta_k^0(t) = \frac{H_e^{[k]}}{C}u(t) + \frac{F_e^{[k]}}{P_d C}y(t) \tag{4.73}$$

The free emulation describes the development of the emulated filtered output δ-derivatives with zero future input. By collecting the emulations for $k = 0, \ldots, N_y$ in a matrix notation we end up with the following vector description of the predictions of the filtered output δ-derivatives

$$\boldsymbol{\zeta}^* = \tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}} + \boldsymbol{\zeta}^0 \tag{4.74}$$

with

$$\boldsymbol{\zeta}^* = \begin{bmatrix} \zeta_0^*(t) \; \zeta_1^*(t) \; \ldots \; \zeta_{N_y}^*(t) \end{bmatrix}^\top \tag{4.75}$$

$$\boldsymbol{\zeta}^0 = \begin{bmatrix} \zeta_0^0(t) \; \zeta_1^0(t) \; \ldots \; \zeta_{N_y}^0(t) \end{bmatrix}^\top \tag{4.76}$$

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} u(t) \; \delta u(t) \; \ldots \; \delta^{\tilde{N}_u} u(t) \end{bmatrix}^\top \tag{4.77}$$

where

$$\tilde{N}_u = N_y + r_p - d \tag{4.78}$$

and row k of $\tilde{\boldsymbol{G}}$ is built up from the coefficients of G_k. The emulator-based predictor can now be written as

$$\begin{aligned}\hat{\zeta}^*(t+\tau) &= \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\boldsymbol{\zeta}^* \\ &= \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\left(\tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}} + \boldsymbol{\zeta}^0\right) \end{aligned} \tag{4.79}$$

As for the unfiltered case we can apply Theorem 11 to get

$$\hat{\zeta}^*(t+kT) = \mathcal{E}\left\{\zeta(t+kT)|\mathcal{I}_{\Delta,t,k}\right\} \tag{4.80}$$

if $N_y \geq k$. That is, the emulation of the filtered system output at the sampling instants coincides with the minimum variance prediction of the filtered output if the order N_y of the emulator-based predictor is chosen high enough.

4.4.3 General constraint on future control input

In the unfiltered versions of the DCGPC and DQGPC we imposed the basic constraint (4.15)

$$\delta^k u(t) = 0 \quad \text{for } k > N_u \tag{4.81}$$

on the future control input where the *control order* N_u satisfies

$$0 \leq N_u \leq \tilde{N}_u \tag{4.82}$$

If, as before, we define

$$\boldsymbol{u} = \left[u(t)\ \delta u(t)\ \delta^2 u(t)\ \ldots\ \delta^{N_u} u(t)\right]^\top \tag{4.83}$$

the emulator-based predictor for the filtered output may be expressed as

$$\hat{\zeta}^*(t+\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\left(\boldsymbol{G}\boldsymbol{u} + \boldsymbol{\zeta}^0\right) \tag{4.84}$$

where $\boldsymbol{G}$ is obtained as the first $N_u + 1$ columns of $\tilde{\boldsymbol{G}}$.

The basic constraint (4.81) is simple and intuitive. To allow more flexibility in the control design we propose a more general constraint on the future control input of the form

$$\boxed{\frac{\phi_n(\delta)}{\phi_d(\delta)} u(t) = 0, \quad N_u < k \le \tilde{N}_u \qquad (4.85)}$$

where

$$\begin{aligned} \deg(\phi_d) &= n_\phi & (4.86) \\ \deg(\phi_n) &= n_\phi + k & (4.87) \end{aligned}$$

i.e. the relative degree of the transfer function $\phi_d(\delta)/\phi_n(\delta)$ is k. As we will see later, this general constraint includes the constraint imposed in the usual q-domain GPC, see Section 4.4.8. Using such a general constraint is a novelty of the described unified framework for discrete-time and continuous-time predictive control.

To incorporate the general constraint into the control design we seek to express the full future control vector $\tilde{\boldsymbol{u}}$ by means of the reduced control vector $\boldsymbol{u}$ and possibly some old input data. This can clearly by done because the constraint for each k relates linearly $u(t + kT)$ to the signals $u(t + (k - 1)T), u(t + (k - 2)T), \ldots$ due to non-causal filter $\phi_n(\delta)/\phi_d(\delta)$. Note, that the following calculations are meaningless in the case $N_u = \tilde{N}_u$ since in that case $\boldsymbol{u} = \tilde{\boldsymbol{u}}$. Hence, in the following assume $N_u < \tilde{N}_u$.

Introduce the Diophantine equation

$$\frac{\phi_n(\delta)}{\phi_d(\delta)} = E^{[k]}(\delta) + \frac{F^{[k]}(\delta)}{\phi_d(\delta)} \qquad (4.88)$$

where $\deg(E^{[k]}) = k$ and $\deg(F^{[k]}) = n_f$ with

$$n_f = \max\{0, n_\phi - 1\} \qquad (4.89)$$

Note, that if $n_\phi = 0$ then $F^{[k]}(\delta) = 0$. Now the constraint (4.85) implies, that

$$E^{[k]}(\delta)u(t) + F^{[k]}(\delta)\frac{u(t)}{\phi_d(\delta)} = 0, \quad N_u < k \leq \tilde{N}_u \tag{4.90}$$

With

$$E^{[k]}(\delta) = e_k^{[k]}\delta^k + \cdots + e_1^{[k]}\delta + e_0^{[k]} \tag{4.91}$$

this can be written as

$$\left(e_k^{[k]}\delta^k + \cdots + e_{N_u+1}^{[k]}\delta^{N_u+1}\right)u(t) + \left(e_{N_u}^{[k]}\delta^{N_u} + \cdots + e_0^{[k]}\right)u(t) \tag{4.92}$$

$$+ F^{[k]}(\delta)\frac{u(t)}{\phi_d(\delta)} = 0, \quad N_u < k \leq \tilde{N}_u \tag{4.93}$$

By putting up this equation for $k = N_u + 1, \ldots, \tilde{N}_u$ we get the following matrix equation

$$\begin{bmatrix} e_{N_u+1}^{[N_u+1]} & 0 & 0 & \cdots & 0 \\ e_{N_u+1}^{[N_u+2]} & e_{N_u+2}^{[N_u+2]} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ e_{N_u+1}^{[\tilde{N}_u-1]} & e_{N_u+2}^{[\tilde{N}_u-1]} & \cdots & e_{\tilde{N}_u-1}^{[\tilde{N}_u-1]} & 0 \\ e_{N_u+1}^{[\tilde{N}_u]} & e_{N_u+2}^{[\tilde{N}_u]} & \cdots & e_{\tilde{N}_u-1}^{[\tilde{N}_u]} & e_{\tilde{N}_u}^{[\tilde{N}_u]} \end{bmatrix} \begin{bmatrix} \delta^{N_u+1}u(t) \\ \vdots \\ \delta^{\tilde{N}_u}u(t) \end{bmatrix} =$$

$$-\begin{bmatrix} e_0^{[N_u+1]} & \cdots & e_{N_u}^{[N_u+1]} \\ \vdots & \ddots & \vdots \\ e_0^{[\tilde{N}_u]} & \cdots & e_{N_u}^{[\tilde{N}_u]} \end{bmatrix} \begin{bmatrix} u(t) \\ \vdots \\ \delta^{N_u}u(t) \end{bmatrix} - \begin{bmatrix} F^{[N_u+1]}(\delta) \\ \vdots \\ F^{[\tilde{N}_u]}(\delta) \end{bmatrix} \frac{u(t)}{\phi_d(\delta)}$$

In a more compact notation we can write

$$\boldsymbol{E}_1 \begin{bmatrix} \delta^{N_u+1}u(t) \\ \vdots \\ \delta^{\tilde{N}_u}u(t) \end{bmatrix} = -\boldsymbol{E}_2\boldsymbol{u} - \boldsymbol{F}\boldsymbol{u}_2 \tag{4.94}$$

where

$$F = \begin{bmatrix} f_{n_f}^{[N_u+1]} & \dots & f_0^{[N_u+1]} \\ \vdots & & \vdots \\ f_{n_f}^{[\tilde{N}_u]} & \dots & f_0^{[\tilde{N}_u]} \end{bmatrix} \tag{4.95}$$

with the coefficients of $F^{[k]}(\delta)$ defined similarly to the coefficients of $E^{[k]}(\delta)$ and

$$\boldsymbol{u}_2 = \begin{bmatrix} \frac{\delta^{n_f} u(t)}{\phi_d} & \frac{\delta^{n_f-1} u(t)}{\phi_d} & \dots & \frac{u(t)}{\phi_d} \end{bmatrix}^\top \tag{4.96}$$

Note, that $\boldsymbol{E}_1$ is lower triangular with non-zero diagonal elements. and hence of full rank. Therefore, we can easily solve for the vector of future input δ-derivatives beyond the control order N_u

$$\begin{bmatrix} \delta^{N_u+1} u(t) \\ \vdots \\ \delta^{\tilde{N}_u} u(t) \end{bmatrix} = -\boldsymbol{E}_1^{-1}\boldsymbol{E}_2\boldsymbol{u} - \boldsymbol{E}_1^{-1}\boldsymbol{F}\boldsymbol{u}_2 \tag{4.97}$$

It easily follows, that the full future control vector $\tilde{\boldsymbol{u}}$ can be expressed as

$$\tilde{\boldsymbol{u}} = \boldsymbol{M}\boldsymbol{u} + \boldsymbol{N}\boldsymbol{u}_2 \tag{4.98}$$

with

$$\boldsymbol{M} = \left[\begin{array}{c} \boldsymbol{I} \\ \hline -\boldsymbol{E}_1^{-1}\boldsymbol{E}_2 \end{array}\right] \tag{4.99}$$

and

$$\boldsymbol{N} = \left[\begin{array}{c} \boldsymbol{0} \\ \hline -\boldsymbol{E}_1^{-1}\boldsymbol{F} \end{array}\right] \tag{4.100}$$

where $\boldsymbol{I}$ and $\boldsymbol{0}$ is an identity matrix and a zero matrix both of dimension $(N_u+1)\times(N_u+1)$.

The dimensions of the involved matrices and vectors are

$$\dim(\boldsymbol{E_1}) = (\tilde{N}_u - N_u)\times(\tilde{N}_u - N_u) \tag{4.101}$$

$$\dim(\boldsymbol{E_2}) = (\tilde{N}_u - N_u)\times(N_u+1) \tag{4.102}$$

$$\begin{aligned}
\dim(\boldsymbol{F}) &= (\tilde{N}_u - N_u) \times (n_f + 1) && (4.103)\\
\dim(\boldsymbol{M}) &= (\tilde{N}_u + 1) \times (N_u + 1) && (4.104)\\
\dim(\boldsymbol{N}) &= (\tilde{N}_u + 1) \times (n_f + 1) && (4.105)\\
\dim(\boldsymbol{u}_2) &= (n_f + 1) \times 1 && (4.106)
\end{aligned}$$

Insertion into (4.74) yields the following emulation of filtered output δ-derivatives subject to the constraint (4.85)

$$\begin{aligned}
\boldsymbol{\zeta}^* &= \tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}} + \boldsymbol{\zeta}^0\\
&= \tilde{\boldsymbol{G}}(\boldsymbol{M}\boldsymbol{u} + \boldsymbol{N}\boldsymbol{u}_2) + \boldsymbol{\zeta}^0\\
&= \tilde{\boldsymbol{G}}\boldsymbol{M}\boldsymbol{u} + \left(\boldsymbol{\zeta}^0 + \tilde{\boldsymbol{G}}\boldsymbol{N}\boldsymbol{u}_2\right) && (4.107)
\end{aligned}$$

I.e. essentially the free emulation is adjusted to take into account the fact, that input δ-derivatives of order higher than N_u possibly depend on past input data. Hence, zeroing the elements of $\boldsymbol{u}$ does not necessarily imply zeroing of all elements of $\tilde{\boldsymbol{u}}$ which is also clear from (4.98). We now obtain the following expression for the emulator-based predictor with the general constraint

$$\hat{\zeta}^*(t+\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\left[\tilde{\boldsymbol{G}}\boldsymbol{M}\boldsymbol{u} + \left(\boldsymbol{\zeta}^0 + \tilde{\boldsymbol{G}}\boldsymbol{N}\boldsymbol{u}_2\right)\right] \qquad (4.108)$$

where $\boldsymbol{G}$ is obtained as the first $N_u + 1$ columns of $\tilde{\boldsymbol{G}}$.

Remark 1: The basic constraint (4.81) is obtained by setting

$$\begin{aligned}
\phi_n(\delta) &= \delta^k && (4.109)\\
\phi_d(\delta) &= 1 && (4.110)
\end{aligned}$$

which results in

$$\boldsymbol{M} = \left[\begin{array}{c} \boldsymbol{I} \\ \hline \boldsymbol{0} \end{array}\right] \qquad (4.111)$$

and

$$\boldsymbol{N} = \left[\begin{array}{c} \boldsymbol{0} \\ \hline \boldsymbol{0} \end{array}\right] \qquad (4.112)$$

Remark 2: Another interesting constraint is

$$\phi_n(\delta) = \delta^k \overline{\phi}_n(\delta) \tag{4.113}$$

$$\phi_d(\delta) = \overline{\phi}_d(\delta) \tag{4.114}$$

where the filter $\overline{\phi}_n(\delta)/\overline{\phi}_d(\delta)$ is exactly proper and could for instance be chosen as $P_n(\delta)/P_d(\delta)$ in case $r_p = 0$. With this choice a filtered control problem may be restated as an unfiltered control problem with a changed noise model. It can then be shown, that the filter $P_d(\delta)/P_n(\delta)$ becomes a factor of the closed-loop transfer functions from noise and input to output, see (Soeterboek 1992).

Remark 3: A third interesting case which will be discussed further in Section 4.4.8 is

$$\phi_n(\delta) = T(1+\delta T)^{k-1}\delta \tag{4.115}$$

$$\phi_d(\delta) = 1 \tag{4.116}$$

This constraint corresponds to the one used in the usual q-domain GPC: The possibly filtered control signal is kept constant beyond the control horizon specified by N_u.

4.4.4 The filtered control input

The filtered input signal is given by

$$u_f(t) = \frac{Q_n(\delta)}{Q_d(\delta)} u(t) \tag{4.117}$$

where $\deg(Q_n) = \deg(Q_d) = n_q$. The filtered input vector is

$$\boldsymbol{u}\mathrm{f} = \left[u_f(t)\ \delta u_f(t)\ \ldots\ \delta^{\tilde{N}_u} u_f(t) \right]^{\mathsf{T}} \tag{4.118}$$

To relate the filtered input vector with the unfiltered input vector consider the Diophantine-equation

$$\delta^i \frac{Q_n}{Q_d} = \Phi_i + \frac{\Omega_i}{Q_d} \tag{4.119}$$

where

$$\begin{aligned} \deg(\Phi_i) &= i &(4.120)\\ \deg(\Omega_i) &= n_\Omega &(4.121) \end{aligned}$$

with

$$n_\Omega = \max\{0, n_q - 1\} \tag{4.122}$$

By means of the above Diophantine-equation we get

$$\delta^i \frac{Q_n}{Q_d} u(t) = \delta^i u_f(t) = \Phi_i u(t) + \Omega_i \frac{u(t)}{Q_d} \tag{4.123}$$

Therefore, when solving the above equation for $i = 0, \ldots, \tilde{N}_u$ and collecting the coefficients of $\Phi_i(\delta)$ and $\Omega_i(\delta)$ in the matrices $\tilde{\boldsymbol{\Phi}}$ and $\boldsymbol{\Omega}$ we can write

$$\boldsymbol{u}\mathrm{f} = \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{u}} + \boldsymbol{\Omega}\boldsymbol{u}_1 \tag{4.124}$$

where $\boldsymbol{u}_1$ is a vector consisting of filtered old input according to

$$\boldsymbol{u}_1 = \begin{bmatrix} \frac{\delta^{n_\Omega} u(t)}{Q_d} & \frac{\delta^{n_\Omega - 1} u(t)}{Q_d} & \ldots & \frac{u(t)}{Q_d} \end{bmatrix}^\top \tag{4.125}$$

Now a Taylor-series approximation of the future input is given as

$$u_f(t,\tau) = \boldsymbol{T}_{\tilde{N}_u}(\tau)\boldsymbol{M}_{s\gamma}(\tilde{N}_u)\boldsymbol{u}\mathrm{f} = \boldsymbol{T}_{\tilde{N}_u}(\tau)\boldsymbol{M}_{s\gamma}(\tilde{N}_u)\left(\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{u}} + \boldsymbol{\Omega}\boldsymbol{u}_1\right) \tag{4.126}$$

With the constraint (4.85) in mind, see (4.98), the filtered control input vector becomes

$$\begin{aligned} \boldsymbol{u}\mathrm{f} &= \tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{u}} + \boldsymbol{\Omega}\boldsymbol{u}_1 \\ &= \tilde{\boldsymbol{\Phi}}(\boldsymbol{M}\boldsymbol{u} + \boldsymbol{N}\boldsymbol{u}_2) + \boldsymbol{\Omega}\boldsymbol{u}_1 \\ &= \tilde{\boldsymbol{\Phi}}\boldsymbol{M}\boldsymbol{u} + \boldsymbol{\Omega}\boldsymbol{u}_1 + \tilde{\boldsymbol{\Phi}}\boldsymbol{N}\boldsymbol{u}_2 &(4.127) \end{aligned}$$

The input approximation now becomes

$$u_f(t,\tau) = \boldsymbol{T}_{\tilde{N}_u}(\tau)\boldsymbol{M}_{s\gamma}(\tilde{N}_u)\left(\tilde{\boldsymbol{\Phi}}\boldsymbol{M}\boldsymbol{u} + \boldsymbol{\Omega}\boldsymbol{u}_1 + \tilde{\boldsymbol{\Phi}}\boldsymbol{N}\boldsymbol{u}_2\right) \tag{4.128}$$

Note, that according to Theorem 10 this approximation is exact at the sampling instants up to time $t + \tilde{N}_u T$. The dimensions of the involved matrices and vectors are

$$\begin{aligned}\dim(\tilde{\boldsymbol{\Phi}}) &= (\tilde{N}_u + 1) \times (\tilde{N}_u + 1) \quad (4.129)\\ \dim(\boldsymbol{\Omega}) &= (\tilde{N}_u + 1) \times (n_\Omega + 1) \quad (4.130)\\ \dim(\boldsymbol{u}_1) &= (n_\Omega + 1) \times 1 \quad (4.131)\end{aligned}$$

4.4.5 The control law

The filtered control criteria (4.56) and (4.57) subject to the general constraint (4.85) can now be stated on unified form as

$$\begin{aligned}J_U &= (\boldsymbol{\zeta}^* - \boldsymbol{w})^\top \boldsymbol{W}_\mathrm{y} (\boldsymbol{\zeta}^* - \boldsymbol{w}) + \rho \boldsymbol{u}_\mathrm{f}{}^\top \boldsymbol{W}_\mathrm{u} \boldsymbol{u}_\mathrm{f}\\ &= \left(\tilde{\boldsymbol{G}}\boldsymbol{M}\boldsymbol{u} + \boldsymbol{\zeta}^0 + \tilde{\boldsymbol{G}}\boldsymbol{N}\boldsymbol{u}_2 - \boldsymbol{w}\right)^\top \boldsymbol{W}_\mathrm{y} \left(\tilde{\boldsymbol{G}}\boldsymbol{M}\boldsymbol{u} + \boldsymbol{\zeta}^0 + \tilde{\boldsymbol{G}}\boldsymbol{N}\boldsymbol{u}_2 - \boldsymbol{w}\right)\\ &\quad + \rho \left(\tilde{\boldsymbol{\Phi}}\boldsymbol{M}\boldsymbol{u} + \boldsymbol{\Omega}\boldsymbol{u}_1 + \tilde{\boldsymbol{\Phi}}\boldsymbol{N}\boldsymbol{u}_2\right)^\top \boldsymbol{W}_\mathrm{u} \left(\tilde{\boldsymbol{\Phi}}\boldsymbol{M}\boldsymbol{u} + \boldsymbol{\Omega}\boldsymbol{u}_1 + \tilde{\boldsymbol{\Phi}}\boldsymbol{N}\boldsymbol{u}_2\right)\end{aligned}$$

where (4.108) and (4.128) have been used and

$$\boldsymbol{W}_\mathrm{y} = \begin{cases} \boldsymbol{M}_{s\gamma}^\top(N_y) \displaystyle\int_{T_1}^{T_2} \boldsymbol{T}_{N_y}^\top(\tau)\boldsymbol{T}_{N_y}(\tau)\, d\tau\ \boldsymbol{M}_{s\gamma}(N_y) \\ \boldsymbol{M}_{s\gamma}^\top(N_y) \displaystyle\sum_{i=N_1}^{N_2} \boldsymbol{T}_{N_y}^\top(iT)\boldsymbol{T}_{N_y}(iT)\boldsymbol{M}_{s\gamma}(N_y) \end{cases} \quad (4.132)$$

$$\boldsymbol{W}_\mathrm{u} = \begin{cases} \boldsymbol{M}_{s\gamma}^\top(\tilde{N}_u) \displaystyle\int_{0}^{T_2-T_1} \boldsymbol{T}_{\tilde{N}_u}^\top(\tau)\boldsymbol{T}_{\tilde{N}_u}(\tau)\, d\tau\ \boldsymbol{M}_{s\gamma}(\tilde{N}_u) \\ \boldsymbol{M}_{s\gamma}^\top(\tilde{N}_u) \displaystyle\sum_{i=0}^{N_2-N_1} \boldsymbol{T}_{\tilde{N}_u}^\top(iT)\boldsymbol{T}_{\tilde{N}_u}(iT)\boldsymbol{M}_{s\gamma}(\tilde{N}_u) \end{cases} \quad (4.133)$$

with topmost entry in $\boldsymbol{W}_\mathrm{y}$ and $\boldsymbol{W}_\mathrm{u}$ belonging to he DCGPC case. The derivative of J_U with respect to $\boldsymbol{u}$ becomes

$$\frac{\partial J_U}{\partial \boldsymbol{u}} = 2\boldsymbol{M}^\top \tilde{\boldsymbol{G}}^\top \boldsymbol{W}_\mathrm{y} \tilde{\boldsymbol{G}}\boldsymbol{M}\boldsymbol{u} + 2\boldsymbol{M}^\top \tilde{\boldsymbol{G}}^\top \boldsymbol{W}_\mathrm{y}(\boldsymbol{\zeta}^0 + \tilde{\boldsymbol{G}}\boldsymbol{N}\boldsymbol{u}_2 - \boldsymbol{w})$$

$$+ 2\rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} \tilde{\boldsymbol{\Phi}} \boldsymbol{M} \boldsymbol{u} + 2\rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} (\boldsymbol{\Omega} \boldsymbol{u}_1 + \tilde{\boldsymbol{\Phi}} \boldsymbol{N} \boldsymbol{u}_2)$$

Setting this derivative equal to zero and solving for $\boldsymbol{u}$ yields the optimal control vector

$$\begin{aligned} \boldsymbol{u} = &-(\boldsymbol{M}^\top \tilde{\boldsymbol{G}}^\top \boldsymbol{W}_\mathrm{y} \tilde{\boldsymbol{G}} \boldsymbol{M} + \rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} \tilde{\boldsymbol{\Phi}} \boldsymbol{M})^{-1} \times \\ &\left[\boldsymbol{M}^\top \tilde{\boldsymbol{G}}^\top \boldsymbol{W}_\mathrm{y} (\boldsymbol{\zeta}^0 + \tilde{\boldsymbol{G}} \boldsymbol{N} \boldsymbol{u}_2 - \boldsymbol{w}) + \rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} (\boldsymbol{\Omega} \boldsymbol{u}_1 + \tilde{\boldsymbol{\Phi}} \boldsymbol{N} \boldsymbol{u}_2)\right] \end{aligned}$$

This can be written more compactly as

$$\boldsymbol{u} = \boldsymbol{K}_1 (\boldsymbol{w} - \boldsymbol{\zeta}^0 - \tilde{\boldsymbol{G}} \boldsymbol{N} \boldsymbol{u}_2) + \boldsymbol{K}_2 \boldsymbol{u}_1 + \boldsymbol{K}_3 \boldsymbol{u}_2 \tag{4.134}$$

where

$$\begin{aligned} \boldsymbol{K}_1 &= \left(\boldsymbol{M}^\top \tilde{\boldsymbol{G}}^\top \boldsymbol{W}_\mathrm{y} \tilde{\boldsymbol{G}} \boldsymbol{M} + \rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} \tilde{\boldsymbol{\Phi}} \boldsymbol{M}\right)^{-1} \boldsymbol{M}^\top \tilde{\boldsymbol{G}}^\top \boldsymbol{W}_\mathrm{y} \\ \boldsymbol{K}_2 &= -\left(\boldsymbol{M}^\top \tilde{\boldsymbol{G}}^\top \boldsymbol{W}_\mathrm{y} \tilde{\boldsymbol{G}} \boldsymbol{M} + \rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} \tilde{\boldsymbol{\Phi}} \boldsymbol{M}\right)^{-1} \rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} \boldsymbol{\Omega} \\ \boldsymbol{K}_3 &= -\left(\boldsymbol{M}^\top \tilde{\boldsymbol{G}}^\top \boldsymbol{W}_\mathrm{y} \tilde{\boldsymbol{G}} \boldsymbol{M} + \rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} \tilde{\boldsymbol{\Phi}} \boldsymbol{M}\right)^{-1} \rho \boldsymbol{M}^\top \tilde{\boldsymbol{\Phi}}^\top \boldsymbol{W}_\mathrm{u} \tilde{\boldsymbol{\Phi}} \boldsymbol{N} \end{aligned}$$

The controllers are implemented following a receding horizon strategy. Hence, only the first element $u(t)$ of the optimal control vector $\boldsymbol{u}$ need to be calculated. Let $\boldsymbol{k}_1^\top$, $\boldsymbol{k}_2^\top$ and $\boldsymbol{k}_3^\top$ denote the first row of $\boldsymbol{K}_1$, $\boldsymbol{K}_2$ and $\boldsymbol{K}_3$, respectively. I.e.

$$\boldsymbol{k}_1^\top = [1\ 0\ \dots\ 0]\, \boldsymbol{K}_1 \tag{4.135}$$

$$\boldsymbol{k}_2^\top = [1\ 0\ \dots\ 0]\, \boldsymbol{K}_2 \tag{4.136}$$

$$\boldsymbol{k}_3^\top = [1\ 0\ \dots\ 0]\, \boldsymbol{K}_3 \tag{4.137}$$

Then the following control signal is applied to the system

$$u(t) = \boldsymbol{k}_1^\top (\boldsymbol{w} - \boldsymbol{\zeta}^0 - \tilde{\boldsymbol{G}} \boldsymbol{N} \boldsymbol{u}_2) + \boldsymbol{k}_2^\top \boldsymbol{u}_1 + \boldsymbol{k}_3^\top \boldsymbol{u}_2 \tag{4.138}$$

4.4.6 Frequency weighted controllers on input-output form

Define the coefficients of $\boldsymbol{k}_1$, $\boldsymbol{k}_2$ and $\boldsymbol{k}_3$ as follows

$$\begin{aligned}
\boldsymbol{k}_1 &= \left[k_{1,0}\ k_{1,1}\ \dots\ k_{1,N_y}\right]^\top && (4.139)\\
\boldsymbol{k}_2 &= \left[k_{2,0}\ k_{2,1}\ \dots\ k_{2,n_\Omega}\right]^\top && (4.140)\\
\boldsymbol{k}_3 &= \left[k_{3,0}\ k_{3,1}\ \dots\ k_{3,n_f}\right]^\top && (4.141)
\end{aligned}$$

In order to get the controllers on input-output form we insert the vector definitions to get

$$\begin{aligned}
u(t) &= \sum_{i=0}^{N_y} k_{1,i}\left(\delta^i w(t) - \frac{H_e^{[i]}}{C}u(t) - \frac{F_e^{[i]}}{P_d C}y(t) - \frac{V_i}{\phi_d}u(t)\right)\\
&\quad + \sum_{i=0}^{n_\Omega} k_{2,i}\delta^{n_\Omega - i}\frac{u(t)}{Q_d} + \sum_{i=0}^{n_f} k_{3,i}\delta^{n_f - i}\frac{u(t)}{\phi_d} \qquad (4.142)
\end{aligned}$$

where

$$V_i(\delta) = [0\ \dots\ 0\ 1\ 0\ \dots\ 0]\,\tilde{\boldsymbol{G}}\boldsymbol{N}\left[\delta^{n_f}\ \delta^{n_f-1}\ \dots\ \delta\ 1\right]^\top \qquad (4.143)$$

with the 1 on the i'th place in the leftmost vector. I.e. $V_i(\delta)$ is determined as the i'th row of the polynomial vector next to the row selection vector. The degree of $V_i(\delta)$ is n_f. The derivation of the input-output form proceeds as follows.

$$\begin{aligned}
P_d C\phi_d Q_d u(t) &= \sum_{i=0}^{N_y} k_{1,i}\Big(P_d\phi_d Q_d C\delta^i w(t) - H_e^{[i]}P_d\phi_d Q_d u(t)\\
&\qquad - F_e^{[i]}\phi_d Q_d y(t) - V_i P_d C Q_d u(t)\Big)\\
&\quad + \sum_{i=0}^{n_\Omega} k_{2,i}P_d C\phi_d\delta^{n_\Omega - i}u(t) + \sum_{i=0}^{n_f} k_{3,i}P_d C Q_d\delta^{n_f - i}u(t)
\end{aligned}$$

which leads to

$$u(t)P_d\Bigg(C\phi_d Q_d + \sum_{i=0}^{N_y} k_{1,i}\left(H_e^{[i]}\phi_d Q_d + V_i C Q_d\right)$$

$$
\left. - \sum_{i=0}^{n_\Omega} k_{2,i} C \phi_d \delta^{n_\Omega - i} - \sum_{i=0}^{n_f} k_{3,i} C Q_d \delta^{n_f - i} \right)
$$

$$
= P_d \phi_d Q_d C \sum_{i=0}^{N_y} k_{1,i} \delta^i w(t) - \sum_{i=0}^{N_y} k_{1,i} F_e^{[i]} \phi_d Q_d y(t)
$$

The RSQ-form is identified with the following polynomials

$$
\begin{aligned}
R(\delta) &= P_d \left(C\phi_d Q_d + \sum_{i=0}^{N_y} k_{1,i} \left(H_e^{[i]} \phi_d Q_d + V_i C Q_d \right) \right. \\
&\qquad \left. - \sum_{i=0}^{n_\Omega} k_{2,i} C \phi_d \delta^{n_\Omega - i} - \sum_{i=0}^{n_f} k_{3,i} C Q_d \delta^{n_f - i} \right) && (4.144) \\
S(\delta) &= \sum_{i=0}^{N_y} k_{1,i} F_e^{[i]} \phi_d Q_d && (4.145) \\
Q(\delta) &= P_d \phi_d Q_d C \sum_{i=0}^{N_y} k_{1,i} \delta^i && (4.146)
\end{aligned}
$$

The degrees of these polynomials are

$$
\begin{aligned}
\deg(R) &= n + n_p + n_q + n_\phi - \Delta && (4.147) \\
\deg(S) &= n + n_p + n_q + n_\phi - 1 && (4.148) \\
\deg(Q) &= n + n_p + n_q + n_\phi + N_y - \Delta && (4.149)
\end{aligned}
$$

Note, that as for the unfiltered controllers, if $\Delta = 0$ then the transfer function $-S(\delta)/R(\delta)$ from $y(t)$ to $u(t)$ is strictly proper. This means, that noise working directly on the system output is not fed directly back to the control input. If $\Delta = 1$ then this transfer function is exactly proper and the latest system output $y(t)$ is utilized in the control input.

4.4.7 Design parameters

A list of additional design parameters introduced in the frequency weighted version of the emulator-based δ-domain GPC algorithms is shown in Ta-

	Description	Restrictions
$P_n(\delta)/P_d(\delta)$	Filter on system output	$\deg(P_n) \geq \deg(P_d)$
$Q_n(\delta)/Q_d(\delta)$	Filter on system input	$\deg(Q_n) = \deg(Q_d)$
$\phi_n(\delta)/\phi_d(\delta)$	Control constraint filter	$\deg(\phi_n) = deg(\phi_d) + k$

Table 4.4: Additional design parameters for frequency weighted version of the DCGPC and DQGPC algorithms.

ble 4.4. These design parameters allow frequency-specific tuning of the controllers as discussed in the beginning of Section 4.4.

4.4.8 Constant future control constraint

In (4.81) we introduced the constraint

$$\delta^k u(t) = 0 \quad \text{for } k > N_u \tag{4.150}$$

in order to reduce the complexity of the minimization problem. This constraint imposes a limitation on the "wildness" of the future control strategy. It is easy to incorporate in the design because it directly zeroes some of the entries in the full future control vector $\tilde{\boldsymbol{u}}$. In this section we will examine an alternative constraint of the future control which resembles the one used in the usual q-domain GPC. Very commonly a constraint like

$$u(t + kT) - u(t + (k-1)T) = 0, \quad k > N_u \tag{4.151}$$

or equivalently

$$q^k(1 - q^{-1})u(t) = 0, \quad k > N_u \tag{4.152}$$

is used in q-domain GPC's. This constraint has the effect of forcing all but the first N_u values of the control signal to be assigned to the value of $u(t + (N_u - 1)T)$, so-called constant future control.

As remarked earlier this can be achieved as a special case of the more general constraint (4.85)

$$\frac{\phi_n(\delta)}{\phi_d(\delta)}u(t) = 0, \quad N_u < k \leq \tilde{N}_u \tag{4.153}$$

where

$$\begin{aligned} \deg(\phi_d) &= n_\phi &(4.154)\\ \deg(\phi_n) &= n_\phi + k &(4.155)\end{aligned}$$

by choosing

$$\begin{aligned} \phi_n(\delta) &= T(1+\delta T)^{k-1}\delta &(4.156)\\ \phi_d(\delta) &= 1 &(4.157)\end{aligned}$$

This follows from the following simple rewritings

$$\begin{aligned} \frac{\phi_n(\delta)}{\phi_d(\delta)} &= q^k(1-q^{-1}) \\ &= (1+\delta T)^k\left(1-\frac{1}{1+\delta T}\right) \\ &= (1+\delta T)^k\frac{\delta}{\delta+\frac{1}{T}} \\ &= T(1+\delta T)^k\delta &(4.158)\end{aligned}$$

Hence, we can by means of the general constraint (4.85) incorporate the usual q-domain GPC in the frequency weighted emulator-based GPC design.

Naturally, this q-domain like constraint on the control vector is not suitable at fast sampling rates. In particular it fails in the continuous-time limit $T \to 0$. The constraint has been introduced in order to allow design of controllers equivalent to the q-domain GPC. In the next section the necessary choice of design parameters to get this equivalence is discussed in more detail.

4.4.9 Connection to basic q-domain GPC

The choice of the general future control constraint described above in combination with a specific choice of filter on the control input allows us to formulate a mathematically equivalent version of the standard GPC in the δ-domain emulator-based GPC framework.

Consider the filtered DQGPC criterion

$$J_Q = \sum_{i=N_1}^{N_2} \left[\hat{y}^*(t+iT) - w(t,iT)\right]^2 + \rho \sum_{i=0}^{N_2-N_1} \left[u_f(t,iT)\right]^2 \tag{4.159}$$

where output filtering has been omitted. The criterion is subject to

$$\frac{\phi_n(\delta)}{\phi_d(\delta)} u(t) = 0, \quad N_u < k \leq \tilde{N}_u \tag{4.160}$$

As shown in Section 4.4.8 the constant future control constraint is obtained by choosing

$$\frac{\phi_n(\delta)}{\phi_d(\delta)} = T(1+\delta T)^{k-1}\delta \tag{4.161}$$

The weighting of control increments in the criterion function can be obtained by choosing the filter

$$\frac{Q_n(\delta)}{Q_d(\delta)} = 1 - q^{-1} = 1 - \frac{1}{1+\delta T} = \frac{\delta}{\delta + \frac{1}{T}} \tag{4.162}$$

To get an exact equivalence to the basic GPC we should use a model with ARIMAX structure. Consider the model

$$A(\delta)y(t) = B(\delta)u(t) + \frac{C(\delta)}{1-q^{-1}} e(t) \tag{4.163}$$

An equivalent ARMAX model is given by

$$A(\delta)\delta y(t) = B(\delta)\delta u(t) + C(\delta)(\delta + \frac{1}{T})e(t) \tag{4.164}$$

where we have inserted $1 - q^{-1} = \delta/(\delta + 1/T)$. Using this model and the design choices given above we achieve with $N_y \geq N_2$ a minimization of the following criterion

$$J = \sum_{i=N_1}^{N_2} \left[\hat{y}(t+iT|\mathcal{I}_{\Delta,t,k}) - w(t+iT)\right]^2 + \rho \sum_{i=0}^{N_u} \left[u(t+iT) - u(t+(i-1)T)\right]^2$$

subject to

$$u(t+kT) = u(t+N_uT), \qquad \text{for } k > N_u \tag{4.165}$$

This is exactly the same criterion minimized by the basic GPC described in Section 4.2.1 except for the information set used in the optimal prediction of the output. With $\deg(C) = \deg(A)$ which is the common choice in discrete-time modelling, the filtered DQGPC is based on incomplete information while the q-domain GPC is based on complete information. This is a slight drawback of the DQGPC approach. However, the DQGPC formulation offers a possible advantage. In the emulator-based design it is possible to obtain computational savings due to the extrapolation facility of the emulator-based predictor, that is choosing $N_y < N_2$.

4.4.10 Considerations for adaptive control

There seems to be a formidable amount of quantities to be computed to derive the weighted δ-domain GPC controllers. However, a lot of these quantities are independent of the model polynomials $A(\delta)$, $B(\delta)$ and $C(\delta)$. Hence, it is sufficient to calculate these quantities only at the initialization of the adaptive loop or whenever the design-variables of the controller are changed.

The matrix $\tilde{\boldsymbol{G}}$ depend on the model polynomials and is found by solving the two coupled Diophantine-equations (4.65) and (4.66) for $k = 0, \ldots, N_y$. This can be done recursively.

With the design parameters of the control criteria fixed, the matrices $\boldsymbol{M}$ and $\boldsymbol{N}$ associated with the general future control constraint are model independent. Also the matrices $\tilde{\boldsymbol{\Phi}}$ and $\boldsymbol{\Omega}$ concerning the filtered control input may be determined without knowledge of the model. Finally, the weighting matrices $\boldsymbol{W}_{\mathrm{y}}$ and $\boldsymbol{W}_{\mathrm{u}}$ in the unified quadratic criteria depend only on the predictor order and the costing horizons.

With the above mentioned matrices in hand, the control law can be determined. From these considerations we observe, that keeping the order of the predictor low reduces the computational complexity of the control design. With a conventional discrete-time predictor the predictor order equals the prediction horizon N_2. Because of the extrapolation facility of the emulator-based predictor the predictor order may in some cases, especially with high values of N_2, be reduced substantially at the cost of small approximation errors. This is very convenient when controlling in an adaptive fashion.

4.5 A state-space approach to emulator-based GPC

This section is concerned with a state-space formulation of the basic DCGPC and DQGPC control designs. Some of the results have previously been presented in (Rostgaard *et al.* 1996) where a simplified version of the DCGPC is considered. The state-space form is attractive from an implementational point of view and more easily extends to the multivariable case as we shall see later. Another advantage, although not treated here, is the possibility to work with a controlled output different from the measured output. Also the state-space system representation gives a freedom to choose a realization appropriate for a given problem, for instance with numerical or computational aspects in mind.

In the state-space formulation the predictive controllers basically reduce to a constant state-feedback combined with a predictive observer and a feedforward from the future reference.

Other work on state-space implementations of predictive control algorithms can be found in (Chisci *et al.* 1996) where the so-called Stabilizing Stochastic Predictive Control (SSPC) algorithm is treated.

4.5.1 The state-space model

We consider a single-input single-output δ-domain state-space model of the form

$$\begin{aligned} \delta x(t) &= \boldsymbol{A}x(t) + \boldsymbol{B}u(t) + v_\delta(t) && (4.166)\\ y(t) &= \boldsymbol{C}x(t) + e_\delta(t) && (4.167) \end{aligned}$$

where $\zeta_{1,t}$ and $\zeta_{2,t}$ are mutually uncorrelated zero-mean noise-processes with variances $\mathcal{E}\{v_\delta(t)v_\delta^\top(t)\} = \boldsymbol{\Omega}_\delta/T$ and $\mathcal{E}\{e_\delta(t)e_\delta^\top(t)\} = \boldsymbol{\Gamma}_\delta/T$, respectively.

The above model corresponds to the choice $\Delta = 0$ for the noise polynomial of the equivalent input-output model. Hence, in this section we will work with incomplete information only, cf. the discussion in Section 3.2.2.

It is convenient to work with the stationary innovations form of the above model. According to Theorem 6, see Section 2.5.6, this is given as

$$\begin{aligned} \delta x(t) &= \boldsymbol{A}x(t) + \boldsymbol{B}u(t) + \boldsymbol{K}e(t) && (4.168)\\ y(t) &= \boldsymbol{C}x(t) + e(t) && (4.169) \end{aligned}$$

where $\{e(t)\}$ is the innovations sequence and $\boldsymbol{K}$ is the Kalman gain (2.116).

4.5.2 Emulator-based prediction in state-space

We want to express the emulations of output δ-derivatives in terms of the state-space model matrices. The second order δ-derivative of the state-vector is given as

$$\begin{aligned}
\delta^2 x(t) &= \boldsymbol{A}\delta x(t) + \boldsymbol{B}\delta u(t) + \boldsymbol{K}\delta e(t) \\
&= \boldsymbol{A}\left(\boldsymbol{A}x(t) + \boldsymbol{B}u(t) + \boldsymbol{K}e(t)\right) + \boldsymbol{B}\delta u(t) + \boldsymbol{K}\delta e(t) \\
&= \boldsymbol{A}^2 x(t) + \boldsymbol{AB}u(t) + \boldsymbol{B}\delta u(t) + \boldsymbol{AK}e(t) + \boldsymbol{K}\delta e(t)
\end{aligned}$$

Following this strategy it is not difficult to see by successive insertion, that the k'th order δ-derivative of the state-vector equals

$$\delta^k x(t) = \boldsymbol{A}^k x(t) + \sum_{i=0}^{k-1} \boldsymbol{A}^{k-i-1}\boldsymbol{B}\delta^i u(t) + \sum_{i=0}^{k-1} \boldsymbol{A}^{k-i-1}\boldsymbol{K}\delta^i e(t) \qquad (4.170)$$

Hence the output δ-derivatives become

$$\begin{aligned}
y_k(t) &= \delta^k y(t) \\
&= \delta^k e(t) + \boldsymbol{CA}^k x(t) + \sum_{i=0}^{k-1} \boldsymbol{CA}^{k-i-1}\boldsymbol{B}\delta^i u(t) + \sum_{i=0}^{k-1} \boldsymbol{CA}^{k-i-1}\boldsymbol{K}\delta^i e(t)
\end{aligned}$$

Define the state estimate obtained by an optimal (minimizing the mean square error) predictive observer as

$$\hat{x}(t) = \mathcal{E}\{x(t)|\mathcal{I}_{i,t}\} \qquad (4.171)$$

Then the emulated output δ-derivatives are given as

$$y_k^*(t) = \mathcal{E}\{y_k(t)|\mathcal{I}_{i,t,k}\} = \boldsymbol{CA}^k \hat{x}(t) + \sum_{i=0}^{k-1} \boldsymbol{CA}^{k-i-1}\boldsymbol{B}\delta^i u(t) \qquad (4.172)$$

By comparing this to the expressions obtained by the input-output approach, see Section 3.4.3, (3.146)

$$y_k^*(t) = G_e^{[k]}(\delta)u(t) + y_k^0(t) \qquad (4.173)$$

we identify the *free emulation* as

$$y_k^0(t) = \boldsymbol{CA}^k \hat{x}(t) = \frac{H_e^{[k]}(\delta)}{C(\delta)}u(t) + \frac{F_e^{[k]}(\delta)}{C(\delta)}y(t) \qquad (4.174)$$

and see, that

$$G_e^{[k]}(\delta) = \sum_{i=0}^{k-1} \boldsymbol{C}\boldsymbol{A}^{k-i-1}\boldsymbol{B}\delta^i \tag{4.175}$$

Define as in the input-output approach

$$\begin{aligned} \boldsymbol{y}^* &= \begin{bmatrix} y_0^*(t) \; y_1^*(t) \; \ldots \; y_{N_y}^*(t) \end{bmatrix}^\top & (4.176) \\ \boldsymbol{y}^0 &= \begin{bmatrix} y_0^0(t) \; y_1^0(t) \; \ldots \; y_{N_y}^0(t) \end{bmatrix}^\top & (4.177) \\ \tilde{\boldsymbol{u}} &= \begin{bmatrix} u(t) \; \delta u(t) \; \ldots \; \delta^{(N_y-1)} u(t) \end{bmatrix}^\top & (4.178) \end{aligned}$$

In a matrix notation we can then express the emulation of output δ-derivatives of order $0, \ldots, N_y$ as

$$\boldsymbol{y}^* = \tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}} + \boldsymbol{y}^0 \tag{4.179}$$

where $\tilde{\boldsymbol{G}}$ is a Toeplitz matrix of dimension $(N_y+1) \times N_y$ built up from what could be called the δ-domain Markov parameters of the state-space model

$$\tilde{\boldsymbol{G}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \boldsymbol{CB} & 0 & \cdots & 0 \\ \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{N_y-1}\boldsymbol{B} & \boldsymbol{CA}^{N_y-2}\boldsymbol{B} & \cdots & \boldsymbol{CB} \end{bmatrix} \tag{4.180}$$

and the vector of free emulations is given as

$$\boldsymbol{y}^0 = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{N_y} \end{bmatrix} \hat{x}(t) = \boldsymbol{\mathcal{O}}\hat{x}(t) \tag{4.181}$$

with $\boldsymbol{\mathcal{O}}$ being an extended observability matrix.

By introducing the usual control constraint

$$\delta^k u(t) = 0, \quad \text{for } k > N_u \tag{4.182}$$

and defining

$$\boldsymbol{u} = \left[u(t)\ \delta u(t)\ \ldots\ \delta^{N_u} u(t)\right]^\top \tag{4.183}$$

we can finally write

$$\boldsymbol{y}^* = \boldsymbol{G}\boldsymbol{u} + \boldsymbol{\mathcal{O}}\hat{x}(t) \tag{4.184}$$

where $\boldsymbol{G}$ is of dimension $(N_y + 1) \times (N_u + 1)$ and is obtained as the first $N_u + 1$ columns of $\tilde{\boldsymbol{G}}$.

The emulator-based predictor can now be expressed as

$$\hat{y}^*(t+\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y)\left(\boldsymbol{G}\boldsymbol{u} + \boldsymbol{\mathcal{O}}\hat{x}(t)\right) \tag{4.185}$$

4.5.3 The state feedback

The solution to the optimal control problem now proceeds exactly as in the input-output formulation. The difference to the input-output approach is the expression of the effect of future control input on the system output δ-derivatives in terms of the state-space matrices and the expression of the free emulation in terms of the state-space matrices and a state estimate.

Hence, the criterion (4.24)

$$J_U = \left(\boldsymbol{G}\boldsymbol{u} + \boldsymbol{y}^0 - \boldsymbol{w}\right)^\top \boldsymbol{W}_{\mathrm{y}} \left(\boldsymbol{G}\boldsymbol{u} + \boldsymbol{y}^0 - \boldsymbol{w}\right) + \rho \boldsymbol{u}^\top \boldsymbol{W}_{\mathrm{u}} \boldsymbol{u} \tag{4.186}$$

gives rise to the optimal future control sequence

$$\boldsymbol{u} = \left(\boldsymbol{G}\boldsymbol{W}_{\mathrm{y}}\boldsymbol{G} + \rho\boldsymbol{W}_{\mathrm{u}}\right)^{-1} \boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}}(\boldsymbol{w} - \boldsymbol{y}^0) \tag{4.187}$$

Define as before $\boldsymbol{k}^\top$ as the first row of the matrix

$$\left(\boldsymbol{G}\boldsymbol{W}_{\mathrm{y}}\boldsymbol{G} + \rho\boldsymbol{W}_{\mathrm{u}}\right)^{-1} \boldsymbol{G}^\top \boldsymbol{W}_{\mathrm{y}} \tag{4.188}$$

Then, using the receding horizon strategy the control law is

$$\begin{aligned} u(t) &= \boldsymbol{k}^\top(\boldsymbol{w} - \boldsymbol{y}^0) \\ &= -\boldsymbol{k}^\top \boldsymbol{\mathcal{O}} \hat{x}(t) + \boldsymbol{k}^\top \boldsymbol{w} \\ &= -\boldsymbol{L}^\top \hat{x}(t) + \boldsymbol{k}^\top \boldsymbol{w} \end{aligned} \tag{4.189}$$

where the state feedback vector is given as

$$\boldsymbol{L} = \boldsymbol{\mathcal{O}}^\top \boldsymbol{k} \tag{4.190}$$

From an implementational point of view the state-space formulation is attractive since we have now in (4.189) expressed the control law as a state feedback from $\hat{x}(t)$ and a feedforward from the reference vector $\boldsymbol{w}$.

4.5.4 The observer dynamics

Consider the stationary predictive observer

$$\delta\hat{x}(t) = \frac{\hat{x}(t+T) - \hat{x}(t)}{T} = \boldsymbol{A}\hat{x}(t) + \boldsymbol{B}u(t) + \boldsymbol{K}\left(y(t) - \boldsymbol{C}\hat{x}(t)\right) \tag{4.191}$$

By choosing $\boldsymbol{K}$ as the Kalman gain (2.106) this observer will in stationarity yield

$$\hat{x}(t) = \mathcal{E}\{x(t)|\mathcal{I}_{i,t}\} \tag{4.192}$$

It is a predictive observer since the sensor deviation $y(t) - \boldsymbol{C}\hat{x}(t)$ is used to correct the estimate at time $t+T$, $\hat{x}(t+T)$. When the above conditional mean is interpreted as wide sense conditional expectation the observer is optimal in a mean-square sense also in the non-Gaussian case, (Middleton and Goodwin 1990). It appears, that this optimal predictive observer is based on incomplete information.

Rearranging (4.191) yields

$$\delta\hat{x}(t) = (\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})\hat{x}(t) + \boldsymbol{B}u(t) + \boldsymbol{K}y(t) \tag{4.193}$$

or

$$\hat{x}(t) = (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1} (\boldsymbol{B}u(t) + \boldsymbol{K}y(t)) \tag{4.194}$$

Combining this with (4.174) gives

$$y_k^0(t) = \boldsymbol{C}\boldsymbol{A}^k (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1} (\boldsymbol{B}u(t) + \boldsymbol{K}y(t)) \tag{4.195}$$

Now

$$\begin{aligned}
\det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C}) &= \det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})(\det(\delta \boldsymbol{I} - \boldsymbol{A}))^{-1} \det(\delta \boldsymbol{I} - \boldsymbol{A}) \\
&= \det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C}) \det((\delta \boldsymbol{I} - \boldsymbol{A})^{-1}) \det(\delta \boldsymbol{I} - \boldsymbol{A}) \\
&= \det(\boldsymbol{I} + \boldsymbol{K}\boldsymbol{C}(\delta \boldsymbol{I} - \boldsymbol{A})^{-1}) \det(\delta \boldsymbol{I} - \boldsymbol{A}) \\
&= \det(1 + \boldsymbol{C}(\delta \boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{K}) \det(\delta \boldsymbol{I} - \boldsymbol{A}) \\
&= \boldsymbol{C}(\delta \boldsymbol{I} - \boldsymbol{A})^{-1} \det(\delta \boldsymbol{I} - \boldsymbol{A})\boldsymbol{K} + \det(\delta \boldsymbol{I} - \boldsymbol{A}) \\
&= \boldsymbol{C} \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A})\boldsymbol{K} + \det(\delta \boldsymbol{I} - \boldsymbol{A}) \\
&= C(\delta) \qquad (4.196)
\end{aligned}$$

An alternative formulation of this results can be found Theorem 20, see Appendix B. From (4.174) it now appears, that

$$\begin{aligned}
H_e^{[k]}(\delta) &= \boldsymbol{C}\boldsymbol{A}^k \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})\boldsymbol{B} \qquad (4.197) \\
F_e^{[k]}(\delta) &= \boldsymbol{C}\boldsymbol{A}^k \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})\boldsymbol{K} \qquad (4.198)
\end{aligned}$$

Hence, a connection between the Diophantine-equation approach from the input-output formulation and the state-space approach has been obtained.

To restate the observer cancellation, see Section 4.3.5, consider the following rewriting of (4.191)

$$\begin{aligned}
\hat{x}(t) &= (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1}(\boldsymbol{B}u(t) + \boldsymbol{K}y(t)) \\
&= (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1}\Big[\boldsymbol{B}u(t) \\
&\qquad + \boldsymbol{K}\left(\boldsymbol{C}(\delta \boldsymbol{I} - \boldsymbol{A})^{-1}(\boldsymbol{B}u(t) + \boldsymbol{K}e(t)) + e(t)\right)\Big] \\
&= (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1}\Big[(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})(\delta \boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}u(t) \\
&\qquad + (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1}(\delta \boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{K}e(t)\Big] \\
&= (\delta \boldsymbol{I} - \boldsymbol{A})^{-1}(\boldsymbol{B}u(t) + \boldsymbol{K}e(t)) \qquad (4.199)
\end{aligned}$$

The assumed stable observer poles which are the zeros of $\det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})$ are cancelled to give the above. Note, that the state estimate obtained in (4.199) is identical to the state estimate in (4.191) only in stationarity as $\hat{x}(t) \to x(t)$. In the transient phase there may be differences due to different initial conditions although the transfer functions are the same.

4.5.5 Closed-loop investigation

Consider the state estimation error

$$\tilde{x}(t) = x(t) - \hat{x}(t) \tag{4.200}$$

By combining the process equation (4.168) and the observer (4.191) this error may be written as

$$\delta \tilde{x}(t) = (\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})\tilde{x}(t) \tag{4.201}$$

The eigenvalues of $\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C}$ are the zeros of $C(\delta)$ and these are assumed to be stable. Hence, the estimation error will tend to zero in stationarity. Inserting the control law (4.189) yields

$$\delta x(t) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{L}^{\top})x(t) + \boldsymbol{B}\boldsymbol{L}^{\top}\tilde{x}(t) + \boldsymbol{B}\boldsymbol{k}^{\top}\boldsymbol{w} + \boldsymbol{K}e(t) \tag{4.202}$$

Hence, as $\tilde{x}(t) \to 0$ we get the following stationary closed-loop relation

$$y(t) = \boldsymbol{C}(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{L}^{\top})^{-1}\left(\boldsymbol{B}\boldsymbol{k}^{\top}\boldsymbol{w} + \boldsymbol{K}e(t)\right) + e(t) \tag{4.203}$$

Again it appears, that the transfer function for the deterministic part of the system from $w(t)$ to $y(t)$ is independent of the observer dynamics while the transfer function from $e(t)$ to $y(t)$ depends on it through $\boldsymbol{K}$.

A further connection to the input-output derivation of the emulator-based GPC algorithms can be established when considering the RSQ-control law

$$R(\delta)u(t) = -S(\delta)y(t) + Q(\delta)w(t) \tag{4.204}$$

To obtain the above form from the state-space derivation consider the observer equation (4.194) and insert this into the control law. Then

$$u(t) = -\boldsymbol{L}^\top (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1} (\boldsymbol{B}u(t) + \boldsymbol{K}y(t)) + \boldsymbol{k}^\top \boldsymbol{w} \tag{4.205}$$

After some rearrangements one gets

$$\begin{aligned}\left(\boldsymbol{I} + \boldsymbol{L}^\top (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1} \boldsymbol{B}\right) u(t) = \\ - \boldsymbol{L}^\top (\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})^{-1} \boldsymbol{K} y(t) + \boldsymbol{k}^\top \boldsymbol{w}\end{aligned} \tag{4.206}$$

Multiplication with $\det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})$ yields

$$\begin{aligned}\left(\det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C}) + \boldsymbol{L}^\top \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})\boldsymbol{B}\right) u(t) = \\ - \boldsymbol{L}^\top \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})\boldsymbol{K} y(t) + \det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})\boldsymbol{k}^\top \boldsymbol{w}\end{aligned} \tag{4.207}$$

By using the algebraic result derived in Appendix B, Theorem 20, see also (4.196), this can be rewritten to

$$\begin{aligned}\det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C} + \boldsymbol{B}\boldsymbol{L}^\top) u(t) = \\ - \boldsymbol{L}^\top \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})\boldsymbol{K} y(t) + \det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})\boldsymbol{k}^\top \boldsymbol{w}\end{aligned} \tag{4.208}$$

Now the controller polynomials in the RSQ form become

$$\begin{aligned} R(\delta) &= \det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C} + \boldsymbol{B}\boldsymbol{L}^\top) \\ S(\delta) &= \boldsymbol{L}^\top \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C})\boldsymbol{K} \\ Q(\delta) &= \det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C}) \sum_{i=0}^{N_y} k_i \delta^i \end{aligned} \tag{4.209}$$

These controller polynomials are of course identical to the ones derived in the input-output case, see (4.36), as the controllers minimize the exact same criteria and the models are input-output equivalent. According to Theorem 21 we can simplify the expression of $S(\delta)$ slightly to

$$S(\delta) = \boldsymbol{L}^\top \operatorname{adj}(\delta \boldsymbol{I} - \boldsymbol{A})\boldsymbol{K} \tag{4.210}$$

4.6 Extension to multivariable systems

One of the advantages of the state-space formulation of the emulator-based GPC controllers is that the derivation easily is extended to the multivariable case. This was briefly mentioned in (Rostgaard *et al.* 1996) and is also remarked in (Chisci *et al.* 1996). A multivariable version of the Continuous-time GPC (MCGPC) has been presented in (Demircioğlu and Gawthrop 1992) where the starting point is a matrix fraction description of the system to be controlled. In (Kinnaert 1989) a multi-variable version of the GPC was presented using polynomial matrices. In this section we will extend the DCGPC and DQGPC to the multi-variable case using a control criterion similar to that used in (Demircioğlu and Clarke 1992) but based on the state-space approach.

We consider two cases for the multivariable extensions. In the first case we generalize to allow individual predictor orders for each system output and individual control orders for each system input in the emulator-based predictor. Furthermore, we also generalize to allow individual design parameters for each input and output in the control criterion. This unfortunately makes the correspondence to the SISO-case a bit less transparent. If we restrict ourselves to one common predictor order for all outputs and one common control order for all inputs and in addition use common design parameters in the control criterion we retain a close correspondence to the SISO-case. This is the second case of the multivariable extensions.

The multivariable model is assumed to be a p-input r-output system of order n on innovations form, i.e.

$$\delta x(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t) + \boldsymbol{K}e(t) \tag{4.211}$$

$$y(t) = \boldsymbol{C}x(t) + e(t) \tag{4.212}$$

where

$$\dim(\boldsymbol{A}) = n \times n \tag{4.213}$$

$$\begin{aligned} \dim(\boldsymbol{B}) &= n \times p & (4.214) \\ \dim(\boldsymbol{C}) &= r \times n & (4.215) \\ \dim(\boldsymbol{K}) &= n \times r & (4.216) \end{aligned}$$

To simplify the notation define in this section

$$\boldsymbol{R}_{N_y}(\tau) = \boldsymbol{T}_{N_y}(\tau)\boldsymbol{M}_{s\gamma}(N_y) \tag{4.217}$$

4.6.1 Generalized extensions

Emulator-based prediction

First consider the case where we allow for different predictor orders N_{yi} for each of the r system outputs and different control orders N_{ui} for each of the p system inputs. Define the vector

$$\overline{\boldsymbol{y}}_i^* = \begin{bmatrix} y_{i0}^*(t) \\ y_{i1}^*(t) \\ \vdots \\ y_{iN_{yi}}^*(t) \end{bmatrix} \quad \text{for } i = 1, \ldots, r \tag{4.218}$$

consisting of the emulations of the first N_{yi} δ-derivatives of the i'th system output

$$y_{ij}^*(t) = \mathcal{E}\{\delta^j y_i(t) | \mathcal{I}_{i,t,j}\} \tag{4.219}$$

Similarly

$$\overline{\boldsymbol{u}}_i = \begin{bmatrix} u_i(t) \\ \delta u_i(t) \\ \vdots \\ \delta^{N_{ui}} u_i(t) \end{bmatrix} \quad \text{for } i = 1, \ldots, p \tag{4.220}$$

is a vector with the first N_{ui} δ-derivatives of the i'th input. Collect these vectors in stacked column vectors

$$\overline{\boldsymbol{y}}^* = \begin{bmatrix} \overline{\boldsymbol{y}}_1^* \\ \overline{\boldsymbol{y}}_2^* \\ \vdots \\ \overline{\boldsymbol{y}}_r^* \end{bmatrix} \qquad \overline{\boldsymbol{u}} = \begin{bmatrix} \overline{\boldsymbol{u}}_1 \\ \overline{\boldsymbol{u}}_2 \\ \vdots \\ \overline{\boldsymbol{u}}_p \end{bmatrix} \tag{4.221}$$

We now wish to establish a matrix expression for the output emulations in terms of the state-space matrices and a state estimate.

Define the column vectors of $\boldsymbol{B}$ as

$$\boldsymbol{B} = [\boldsymbol{B}_1 \ \boldsymbol{B}_2 \ \ldots \ \boldsymbol{B}_p] \tag{4.222}$$

and the row vectors of $\boldsymbol{C}$ as

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_1 \\ \boldsymbol{C}_2 \\ \vdots \\ \boldsymbol{C}_r \end{bmatrix} \tag{4.223}$$

where $\boldsymbol{B}_i$ corresponds to the i'th input and $\boldsymbol{C}_i$ corresponds to the i'th output. Let

$$\overline{\boldsymbol{G}}_{ij} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \boldsymbol{C}_i\boldsymbol{B}_j & 0 & \cdots & 0 \\ \boldsymbol{C}_i\boldsymbol{A}\boldsymbol{B}_j & \boldsymbol{C}_i\boldsymbol{B}_j & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_i\boldsymbol{A}^{N_{uj}}\boldsymbol{B}_j & \boldsymbol{C}_i\boldsymbol{A}^{N_{uj}-1}\boldsymbol{B}_j & \cdots & \boldsymbol{C}_i\boldsymbol{B}_j \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}_i\boldsymbol{A}^{N_{yi}-1}\boldsymbol{B}_j & \boldsymbol{C}_i\boldsymbol{A}^{N_{yi}-2}\boldsymbol{B}_j & \cdots & \boldsymbol{C}_i\boldsymbol{A}^{N_{yi}-N_{uj}-1}\boldsymbol{B}_j \end{bmatrix} \tag{4.224}$$

and

$$\overline{\mathcal{O}}_i = \begin{bmatrix} \boldsymbol{C}_i \\ \boldsymbol{C}_i\boldsymbol{A} \\ \boldsymbol{C}_i\boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{C}_i\boldsymbol{A}^{N_{yi}} \end{bmatrix} \tag{4.225}$$

With these definitions it is not difficult to see, that

$$\overline{\boldsymbol{y}}^* = \begin{bmatrix} \overline{\boldsymbol{G}}_{11} & \cdots & \overline{\boldsymbol{G}}_{1p} \\ \vdots & \ddots & \vdots \\ \overline{\boldsymbol{G}}_{r1} & \cdots & \overline{\boldsymbol{G}}_{rp} \end{bmatrix} \overline{\boldsymbol{u}} + \begin{bmatrix} \overline{\mathcal{O}}_1 \\ \vdots \\ \overline{\mathcal{O}}_r \end{bmatrix} \hat{x}(t) \tag{4.226}$$

or more compact

$$\overline{\boldsymbol{y}}^* = \overline{\boldsymbol{G}}\,\overline{\boldsymbol{u}} + \overline{\mathcal{O}}\,\hat{x}(t) \tag{4.227}$$

The output prediction is obtained as

$$\begin{aligned} \hat{y}^*(t+\tau) &= \begin{bmatrix} \boldsymbol{R}_{N_{y1}}(\tau)\overline{\boldsymbol{y}}_1^* \\ \vdots \\ \boldsymbol{R}_{N_{yr}}(\tau)\overline{\boldsymbol{y}}_r^* \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}_{N_{y1}}(\tau) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{R}_{N_{yr}}(\tau) \end{bmatrix} \overline{\boldsymbol{y}}^* \end{aligned}$$

Multivariable control criteria

Consider the following control criteria

$$\begin{aligned} J_{MC} &= \sum_{i=1}^{r} \lambda_i \int_{T_{1i}}^{T_{2i}} \left[\hat{y}_i^*(t+\tau) - w_i(t,\tau)\right]^2 \, d\tau + \sum_{i=1}^{r} \rho_i \int_0^{T_{ui}} \left[u_i(t,\tau)\right]^2 \, d\tau \\ J_{MQ} &= \sum_{i=1}^{p} \lambda_i \sum_{i=N_{1i}}^{N_{2i}} \left[\hat{y}_i^*(t+iT) - w_i(t,iT)\right]^2 + \sum_{i=1}^{r} \rho_i \sum_{i=0}^{N_{ui}} \left[u_i(t,iT)\right]^2 \end{aligned}$$

These criteria are straightforward extensions of the SISO criteria. We simply add up the contributions for each output and input. There is no difficulty in also allowing crossweighting between any two inputs or any two outputs. However, we omit this for simplicity. The integral-based criterion J_{MC} is very similar to the criterion used in the MCGPC in (Demircioğlu and Gawthrop 1992).

Note, that we have allowed for individual weighting of the system outputs and inputs through the scalar weighting parameters λ_i and ρ_i. The output scaling parameters are a bit redundant as individual output weighting can be achieved to a certain extent by selecting different prediction horizons T_{2i} for the outputs.

The optimal control law

In the following derivation we consider the continuous-time like criterion J_{MC}. Approximate the future input signal as

$$u(t,\tau) = \begin{bmatrix} \boldsymbol{R}_{N_{u1}}(\tau) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{R}_{N_{up}}(\tau) \end{bmatrix} \overline{\boldsymbol{u}} \tag{4.228}$$

where

$$u(t,\tau) = \begin{bmatrix} u_1(t,\tau) \\ u_2(t,\tau) \\ \vdots \\ u_p(t,\tau) \end{bmatrix} \tag{4.229}$$

Similarly, the future reference signal is approximated as

$$w(t,\tau) = \begin{bmatrix} \boldsymbol{R}_{N_{y1}}(\tau) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{R}_{N_{yr}}(\tau) \end{bmatrix} \overline{\boldsymbol{w}} \tag{4.230}$$

where $\overline{\boldsymbol{w}}$ is defined as

$$\overline{\boldsymbol{w}} = \begin{bmatrix} \overline{\boldsymbol{w}}_1 \\ \overline{\boldsymbol{w}}_2 \\ \vdots \\ \overline{\boldsymbol{w}}_p \end{bmatrix} \tag{4.231}$$

with

$$\overline{\boldsymbol{w}}_i = \begin{bmatrix} w_i(t) \\ \delta w_i(t) \\ \vdots \\ \delta^{N_{yi}} w_i(t) \end{bmatrix} \qquad \text{for } i = 1, \ldots, r \tag{4.232}$$

Focusing on the i'th term of the first sum in the control criterion we get

$$\begin{aligned} & \lambda_i \int_{T_{1i}}^{T_{2i}} \left[\hat{y}_i^*(t+\tau) - w_i(t,\tau)\right]^2 \, d\tau \\ = \; & \lambda_i \int_{T_{1i}}^{T_{2i}} \left[\boldsymbol{R}_{N_{yi}}(\tau)(\overline{\boldsymbol{y}}_i^* - \overline{\boldsymbol{w}}_i)\right]^2 \, d\tau \\ = \; & \lambda_i (\overline{\boldsymbol{y}}_i^* - \overline{\boldsymbol{w}}_i)^\top \int_{T_{1i}}^{T_{2i}} \boldsymbol{R}_{N_{yi}}^\top(\tau) \boldsymbol{R}_{N_{yi}}(\tau) \, d\tau (\overline{\boldsymbol{y}}_i^* - \overline{\boldsymbol{w}}_i) \\ = \; & (\overline{\boldsymbol{y}}_i^* - \overline{\boldsymbol{w}}_i)^\top \boldsymbol{R}_{yi} (\overline{\boldsymbol{y}}_i^* - \overline{\boldsymbol{w}}_i) \end{aligned} \tag{4.233}$$

where

$$\boldsymbol{R}_{yi} = \lambda_i \int_{T_{1i}}^{T_{2i}} \boldsymbol{R}_{N_{yi}}^\top(\tau) \boldsymbol{R}_{N_{yi}}(\tau) \, d\tau \tag{4.234}$$

By making the above rewritings for all the terms in the control criterion we end up with the following matrix form

$$J_{MC} = (\overline{\boldsymbol{y}}^* - \overline{\boldsymbol{w}})^\top \boldsymbol{R}_y (\overline{\boldsymbol{y}}^* - \overline{\boldsymbol{w}}) + \overline{\boldsymbol{u}}^\top \boldsymbol{R}_u \overline{\boldsymbol{u}} \tag{4.235}$$

where

$$\boldsymbol{R}_y = \begin{bmatrix} \boldsymbol{R}_{y1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{R}_{yr} \end{bmatrix} \tag{4.236}$$

and

$$\boldsymbol{R}_u = \begin{bmatrix} \boldsymbol{R}_{u1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{R}_{up} \end{bmatrix} \tag{4.237}$$

with

$$\boldsymbol{R}_{ui} = \rho_i \int_0^{T_{ui}} \boldsymbol{R}_{N_{ui}}^\top(\tau)\boldsymbol{R}_{N_{ui}}(\tau)\, d\tau \tag{4.238}$$

It appears, that crossweighting between any two outputs or any two inputs can be obtained by introducing non-zero off-diagonal block elements in $\boldsymbol{R}_y$ or $\boldsymbol{R}_u$.

The optimal future control vector is now obtained by inserting (4.227), finding the gradient with respect to $\overline{\boldsymbol{u}}$, setting this to zero and solving for $\overline{\boldsymbol{u}}$. This results in

$$\begin{aligned} \overline{\boldsymbol{u}} &= (\overline{\boldsymbol{G}}^\top \boldsymbol{R}_y \overline{\boldsymbol{G}} + \boldsymbol{R}_u)^{-1} \overline{\boldsymbol{G}}^\top \boldsymbol{R}_y (\overline{\boldsymbol{w}} - \overline{\boldsymbol{\mathcal{O}}}\hat{x}(t)) \\ &= \overline{\boldsymbol{K}}(\overline{\boldsymbol{w}} - \overline{\boldsymbol{\mathcal{O}}}\hat{x}(t)) \end{aligned} \tag{4.239}$$

where

$$\overline{\boldsymbol{K}} = (\overline{\boldsymbol{G}}^\top \boldsymbol{R}_y \overline{\boldsymbol{G}} + \boldsymbol{R}_u)^{-1} \overline{\boldsymbol{G}}^\top \boldsymbol{R}_y \tag{4.240}$$

As in the SISO-case the controller is meant to be implemented using the receding horizon principle. Hence, pick out the first element of $\overline{\boldsymbol{u}}_1,\ldots,\overline{\boldsymbol{u}}_p$ and apply these to the system. That is, collect the rows of $\overline{\boldsymbol{K}}$ corresponding to the inputs $u_1(t),\ldots,u_p(t)$ in the matrix $\overline{\boldsymbol{k}}$. Then

$$u(t) = \overline{\boldsymbol{k}}(\overline{\boldsymbol{w}} - \overline{\boldsymbol{\mathcal{O}}}\hat{x}(t)) \tag{4.241}$$

where the state estimate is obtained by means of a predictive observer exactly as in the SISO-case.

It is immediate to see, that the summation based criterion J_{MQ} is handled analogously. In this case simply set

$$\boldsymbol{R}_{yi} = \lambda_i \sum_{i=N_{1i}}^{N_{2i}} \boldsymbol{R}_{N_{yi}}^\top(iT)\boldsymbol{R}_{N_{yi}}(iT) \tag{4.242}$$

$$\boldsymbol{R}_{ui} \;=\; \rho_i \sum_{i=0}^{N_{ui}} \boldsymbol{R}_{N_{ui}}^{\top}(iT)\boldsymbol{R}_{N_{ui}}(iT) \tag{4.243}$$

4.6.2 Simplified extensions

Emulator-based prediction

Consider now the case of common predictor and control orders for each of the inputs and outputs, that is

$$N_{yi} \;=\; N_y, \quad \text{for } i = 1, \ldots, r \tag{4.244}$$

$$N_{ui} \;=\; N_u, \quad \text{for } i = 1, \ldots, p \tag{4.245}$$

By rearranging the order of entries in the stacked input and output vectors we can derive much simpler expressions for the emulator-based predictor. Let

$$\overline{\overline{\boldsymbol{y}}}_i^* = \begin{bmatrix} y_{1i}^*(t) \\ y_{2i}^*(t) \\ \vdots \\ y_{ri}^*(t) \end{bmatrix} \qquad \text{for } i = 0, \ldots, N_y \tag{4.246}$$

This vector consists of emulations of the i'th order δ-derivatives of all r system outputs. Define similarly

$$\overline{\overline{\boldsymbol{u}}}_i = \begin{bmatrix} \delta^i u_1(t) \\ \delta^i u_2(t) \\ \vdots \\ \delta^i u_p(t) \end{bmatrix} \qquad \text{for } i = 0, \ldots, N_u \tag{4.247}$$

as a vector containing the i'th order δ-derivatives of all p system inputs.

Define the stacked column vectors

$$\overline{\overline{\boldsymbol{y}}}^* = \begin{bmatrix} \overline{\overline{\boldsymbol{y}}}_0^* \\ \overline{\overline{\boldsymbol{y}}}_1^* \\ \vdots \\ \overline{\overline{\boldsymbol{y}}}_{N_y}^* \end{bmatrix} \qquad \overline{\overline{\boldsymbol{u}}} = \begin{bmatrix} \overline{\overline{\boldsymbol{u}}}_0 \\ \overline{\overline{\boldsymbol{u}}}_1 \\ \vdots \\ \overline{\overline{\boldsymbol{u}}}_{N_u} \end{bmatrix} \tag{4.248}$$

and the matrices

$$\overline{\overline{\boldsymbol{G}}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \boldsymbol{CB} & 0 & \cdots & 0 \\ \boldsymbol{CAB} & \boldsymbol{CB} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{N_u}\boldsymbol{B} & \boldsymbol{CA}^{N_u-1}\boldsymbol{B} & \cdots & \boldsymbol{CB} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{N_y-1}\boldsymbol{B} & \boldsymbol{CA}^{N_y-2}\boldsymbol{B} & \cdots & \boldsymbol{CA}^{N_y-N_u-1}\boldsymbol{B} \end{bmatrix} \tag{4.249}$$

and

$$\overline{\overline{\boldsymbol{\mathcal{O}}}} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{N_y} \end{bmatrix} \tag{4.250}$$

It now holds, that

$$\overline{\overline{\boldsymbol{y}}}^* = \overline{\overline{\boldsymbol{G}}}\,\overline{\overline{\boldsymbol{u}}} + \overline{\overline{\boldsymbol{\mathcal{O}}}}\,\hat{x}(t) \tag{4.251}$$

Note, that the matrices $\overline{\overline{\boldsymbol{G}}}$ and $\overline{\overline{\boldsymbol{\mathcal{O}}}}$ are defined exactly as in the SISO-case but using the system matrices from the MIMO-model. Hence, the simplified multivariable extensions give a much closer computational correspondence to the SISO-case than the general extensions.

The output prediction is obtained as

$$\hat{y}^*(t+\tau) = \overline{\overline{\boldsymbol{R}}}_{N_y}(\tau)\,\overline{\overline{\boldsymbol{y}}}^* \tag{4.252}$$

with

$$\overline{\overline{\boldsymbol{R}}}_{N_y}(\tau) = \boldsymbol{R}_{N_y}(\tau) \otimes \boldsymbol{I}_r \tag{4.253}$$

with $\otimes$ being the Kronecker product and $\boldsymbol{I}_r$ an identity matrix of dimension $r \times r$.

Multivariable control criterion

Consider the following simplified versions of the control criteria

$$\begin{aligned}
J_{MC} &= \sum_{i=1}^{r} \int_{T_1}^{T_2} \left[\hat{y}_i^*(t+\tau) - w_i(t,\tau)\right]^2 \, d\tau + \sum_{i=1}^{r} \rho \int_{0}^{T_u} \left[u_i(t,\tau)\right]^2 \, d\tau \\
J_{MQ} &= \sum_{i=1}^{p} \sum_{i=N_1}^{N_2} \left[\hat{y}_i^*(t+iT) - w_i(t,iT)\right]^2 \; + \sum_{i=1}^{r} \rho \sum_{i=0}^{N_u} \left[u_i(t,iT)\right]^2
\end{aligned}$$

Apart from assuming common predictor and control orders we now also restrict ourselves to common control horizons for the inputs and for the outputs, that is

$$\begin{aligned}
T_{1i} &= T_1, \quad \text{for } i = 1, \ldots, r &(4.254)\\
T_{2i} &= T_2, \quad \text{for } i = 1, \ldots, r &(4.255)\\
T_{ui} &= T_u, \quad \text{for } i = 1, \ldots, p &(4.256)
\end{aligned}$$

or

$$\begin{aligned}
N_{1i} &= N_1, \quad \text{for } i = 1, \ldots, r &(4.257)\\
N_{2i} &= N_2, \quad \text{for } i = 1, \ldots, r &(4.258)\\
N_{ui} &= N_u, \quad \text{for } i = 1, \ldots, p &(4.259)
\end{aligned}$$

Also, we use common control and output weights. This of course makes the output weight superfluous.

As before the following derivations are concerned with J_{MC}. The future input signal can be approximated as

$$u(t,\tau) = \overline{\overline{\boldsymbol{R}}}_{N_u}(\tau)\,\overline{\overline{\boldsymbol{u}}} \tag{4.260}$$

with

$$\overline{\overline{\boldsymbol{R}}}_{N_u}(\tau) = \boldsymbol{R}_{N_u}(\tau) \otimes \boldsymbol{I}_p \tag{4.261}$$

and the future reference as

$$w(t,\tau) = \overline{\overline{\boldsymbol{R}}}_{N_y}(\tau)\,\overline{\overline{\boldsymbol{w}}} \tag{4.262}$$

where

$$\overline{\overline{\boldsymbol{w}}} = \begin{bmatrix} \overline{\overline{\boldsymbol{w}}}_0 \\ \overline{\overline{\boldsymbol{w}}}_1 \\ \vdots \\ \overline{\overline{\boldsymbol{w}}}_{N_y} \end{bmatrix} \tag{4.263}$$

with

$$\overline{\overline{\boldsymbol{w}}}_i = \begin{bmatrix} \delta^i w_1(t) \\ \delta^i w_2(t) \\ \vdots \\ \delta^i w_r(t) \end{bmatrix} \qquad \text{for } i = 1, \ldots, N_y \tag{4.264}$$

Define

$$\boldsymbol{R}_y = \int_{T_1}^{T_2} \boldsymbol{R}_{N_y}(\tau)^\top \boldsymbol{R}_{N_y}(\tau)\, d\tau \tag{4.265}$$

and

$$\boldsymbol{R}_u = \rho \int_0^{T_u} \boldsymbol{R}_{N_u}(\tau)^\top \boldsymbol{R}_{N_u}(\tau)\, d\tau \tag{4.266}$$

Then it is straightforward to see, that the control criterion can be stated on matrix form as

$$J_{MC} = (\overline{\overline{\boldsymbol{y}}}^* - \overline{\overline{\boldsymbol{w}}})^\top \overline{\overline{\boldsymbol{R}}}_y (\overline{\overline{\boldsymbol{y}}}^* - \overline{\overline{\boldsymbol{w}}}) + \overline{\overline{\boldsymbol{u}}}^\top \overline{\overline{\boldsymbol{R}}}_u \overline{\overline{\boldsymbol{u}}} \tag{4.267}$$

with

$$\overline{\overline{\boldsymbol{R}}}_y = \boldsymbol{R}_y \otimes \boldsymbol{I}_r \tag{4.268}$$

$$\overline{\overline{\boldsymbol{R}}}_u = \boldsymbol{R}_u \otimes \boldsymbol{I}_p \tag{4.269}$$

The criterion function can now be minimized exactly as in the case of the general extensions. The applied control input is the optimal value of $\overline{\overline{\boldsymbol{u}}}_0$.

Again, it is immediate, that the summation based criterion is handled very similarly. Simply set

$$\boldsymbol{R}_y = \sum_{i=N_1}^{N_2} \boldsymbol{R}_{N_y}^{\top}(iT)\boldsymbol{R}_{N_y}(iT) \tag{4.270}$$

$$\boldsymbol{R}_u = \rho \sum_{i=0}^{N_u} \boldsymbol{R}_{N_u}^{\top}(iT)\boldsymbol{R}_{N_u}(iT) \tag{4.271}$$

4.7 Chapter conclusion

New predictive control algorithms formulated using discrete-time δ-operator models have been presented in a unified manner. The DCGPC controller is a close approximation of the continuous-time GPC at reasonable fast sampling rates and has a well-defined continuous-time limit. The DQGPC controller has been shown to be closely related to usual q-domain GPC based on optimal prediction. The DQGPC algorithm potentially offers computational savings if the maximum cost horizon N_2 is large at the cost of very small deviations from the optimal solution.

A novelty of the described approaches is, that we in the DCGPC consider a continuous-time like integral-based criterion function but use discrete-time δ-domain models. The possibility to do this is due to emulator-based predictor presented in Chapter 3. This gives a control criterion which is insensitive to the choice of sampling period and which has a well-defined limit in the continuous-time case. This is illustrated by simulation examples. Thus, we have developed a unified framework for discrete-time and continuous-time predictive control algorithms.

It has been shown how general frequency weighted extensions to these two basic algorithms can be incorporated in the design. A special case of the frequency weighted algorithms is a controller almost exactly equivalent to the usual q-domain GPC. This is obtained by introducing a new general constraint on the future input.

A state-space formulation of the basic δ-domain GPC algorithms has been presented. The state-space version is convenient for implementation and is more easily extended to the multivariable case. This has been done in the last section of the chapter.

Chapter 5

Connecting Identification and Control

5.1 Introduction

It is impossible to avoid errors when modelling physical systems. With low-complexity models the modelling error may result in a deterioration of closed-loop performance unless special counter-measures are taken. The main subject of this chapter is the investigation of possible methods to improve the connection between system identification and control design in view of modelling errors..

Traditionally system identification and control design have been considered separate disciplines. Most often they have been combined by *ad hoc* approaches, for instance using the certainty equivalence principle: Find a model estimate and design the controller as if it was the true system. In addition, model estimation is often considered an open-loop problem. As pointed out by (Skelton 1989) there are several difficulties involved in not paying regard to the dependence between the modelling problem and the control design. He illustrates a number of so-called *modelling principles* of which the following are of interest here:

1. Arbitrarily small modelling errors can lead to arbitrarily bad closed-loop behaviour.
2. Large open-loop modelling errors do not necessarily lead to bad closed-loop performance.
3. Open-loop modelling errors (and hence their bounds) do not generally constitute enough information for successful control design.
4. For every model of the system, there exists a model error vector that will correct the model output response. This model error vector always decomposes into four parts: terms due to parameter errors; errors in model order; neglected disturbances; and non-linearities.
5. The modelling and control problems are not separable and are necessarily iterative.

Modelling principle two and three imply, that modelling errors do not need to be small but simply appropriate for control design. Hence, one should not necessarily aim for a model which gives a good description of the system. This can be illustrated by a very simple example, see (Skelton 1989) for more details.

Example 8 *Consider a continuous-time system described by the transfer function*

$$G_0(s) = \frac{1}{(s+1)(0.01s+1)^2} \tag{5.1}$$

and modelled by

$$\hat{G}(s) = \frac{1}{s+1} \tag{5.2}$$

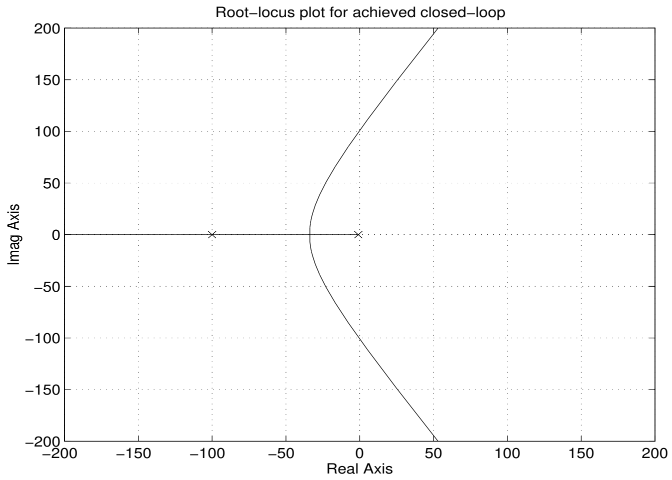

Figure 5.1: Root-locus plot for achieved closed-loop of Example 8.

This is not an unlikely situation in practice, as fast parasitic effects are often ignored. The open-loop step-responses of these two models are very close as is the amplitude spectrum at low frequencies. However, with output feedback the validity of the model $\hat{G}(s)$ depends on the controller as illustrated next. Consider a constant output feedback $u(t) = -Cy(t)$. The achieved *closed-loop system transfer function becomes*

$$T_0(s) = \frac{G_0(s)}{1 + G_0(s)C} = \frac{1}{(s+1)(0.01s+1)^2 + C} \tag{5.3}$$

while the designed closed-loop transfer function is

$$T(s) = \frac{\hat{G}(s)}{1 + \hat{G}(s)C} = \frac{1}{s + (1 + C)} \tag{5.4}$$

The pole of the designed closed-loop system is $s = -(C+1)$ which is stable for all $C > -1$. The poles of the achieved closed-loop system are depicted in the root-locus plot shown in Figure 5.1.

For small values of C the poles due to the ignored dynamics cluster around *the point* $(-100, 0)$ *and do not influence the closed-loop responses significantly. However, it appears that for some* C *(*≈ 204*) the achieved closed-loop becomes unstable. Although this is not surprising the example shows, that the usefulness of the model* $\hat{G}(s)$ *depends both on the modelling error and the controller. We conclude, that the modelling and control problems should not be treated independently.* □

Modelling principle four tells us, that in real life modelling errors are unavoidable. Combined with modelling principle one we may conclude, that system identification and/or control design should somehow take into account the modelling error to achieve acceptable closed-loop behaviour. Furthermore, the identified model should not be thought of as a model the plant; we should rather consider the usefulness of the interaction between model and controller for a given true plant.

There are two main approaches in system identification to improve the interconnection with control design (Van den Hof and Schrama 1995).

1. Quantification of model error.
2. Control-relevant identification.

We now give a brief description of the two approaches.

Quantification of model error

System identification algorithms do not have as a natural part a description of the estimation error. If the model set includes the true system then variance-expressions for parameter estimates exist, see (Ljung 1987). However, low complexity models will inevitable leads to discrepancies between the estimated model and the real system, i.e. unmodelled dynamics. The estimation error will then consist of both variance errors and bias errors. An often used approach in system identification is to evaluate the suitability of the estimated model by model validation techniques. However, in model-based control applications it is difficult to validate the model without actually applying the resulting controller to the real system, cf. Example 8. This is unfortunate if the control design does not take into account knowledge of the estimation error. One way to quantify the estimation error has been presented in (Goodwin and Salgado 1989, Goodwin *et al.* 1992), the so-called stochastic embedding approach. The idea is to model the uncertainties as stochastic variables with a parametric probability distribution. The parameters of this along with a nominal model are then estimated based on measurements. By finding confidence-intervals for the stochastic description of the model errors a frequency-dependent bound is generated. This bound can then be utilized for instance in a subsequent frequency weighted $\mathcal{H}_\infty$ robust control design, (Zhou *et al.* 1996). One issue in such an interconnection is the discussion of hard versus soft error bounds. The quantification of the estimation error results in soft bounds, i.e. confidence intervals, while robust control design methods usually require hard bounds, i.e. guaranteed bounds on model errors. A further discussion can be found in (Gevers 1991).

Control-relevant identification

The second approach focuses on the nominal model. By matching the identification and control design problems a nominal model suitable for high performance control may be obtained. Several approaches where the identification criterion is derived directly from the control criterion giving truly control-relevant identification have been proposed. One difficulty is the fact, that this often results in nonstandard identification problems for which new techniques must be developed. In this chapter we concentrate on this second approach.

Skeltons modelling principle five has proven to be an important observation, as later pinpointed by (Schrama 1992). A number of iterative schemes for improvement of closed-loop performance have been proposed. The idea is to use successive steps of local control design with fixed model and local identification problems with fixed controller. Adaptive control may be considered a special case of the iterative approach. However, it is not easy to treat the bias aspect in adaptive algorithms for two reasons. Firstly, the bias is a property of the asymptotic model estimate. Secondly, the interaction between control design and model estimate is much more difficult to analyze using a controller with fast time-variations, e.g. stability and robustness analysis with modelling errors. The main part of the chapter is concerned with iterative approaches, where the controller is time-invariant during each iteration.

The outline of the chapter is as follows. First some basic definitions concerning the system identification setup are given. The concept of bias is introduced. The asymptotic frequency distribution of bias is analyzed, both for open-loop and closed-loop systems. This analysis is used to find control-relevant prefilters in estimation. Subsequently, the idea of iterative identification and control is motivated and some of the currently prevailing methods are reviewed. Finally, new combined adaptive/iterative approaches are put forward and a conclusion is given.

5.2 System identification

The system identification setup we consider is adopted from (Ljung 1987). We start by giving some basic definitions.

5.2.1 Quasi-stationary signals

Definition 1 (Ljung 1987)

Consider a signal sequence $\{s(t)\}$ subject to the following assumptions

$$\begin{aligned} \mathcal{E}\{s(t)\} &= m_s(t), \quad |m_s(t)| \le C, \quad \forall t \\ \mathcal{E}\{s(t)s(r)\} &= R_s(t,r), \quad |R_s(t,r)| \le C, \end{aligned} \tag{5.5}$$

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} R_s(t, t-\tau) = R_s(\tau), \quad \forall \tau \tag{5.6}$$

where C is a constant. If these assumptions hold then we say, that the process $\{s(t)\}$ is *quasi-stationary.* The function $R_s(\tau)$ is called the covariance function of $s(t)$.

Two signals $\{s(t)\}$ and $\{w(t)\}$ are jointly quasi-stationary if they are both quasi-stationary and the cross-covariance function

$$R_{sw}(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \mathcal{E}\{s(t)w(t-\tau)\} \tag{5.7}$$

exists. Two jointly quasi-stationary signals are uncorrelated if this cross-covariance function is identically zero. □

The reason for introducing the concept of quasi-stationarity is that we often consider signals which have both stochastic and deterministic components. For instance, consider the signal

$$y(t) = G(q)u(t) + H(q)e(t) \tag{5.8}$$

where $e(t)$ is white noise and $u(t)$ is a deterministic input signal. The signal $y(t)$ is not stationary as $\mathcal{E}\{y(t)\} = G(q)u(t)$ is not time-invariant. However, it is a quasi-stationary signal.

5.2.2 Signal spectra

Definition 2 (Ljung 1987)

The power spectrum of the quasi-stationary signal $s(t)$ is defined as

$$\Phi_s(\omega) = \sum_{\tau=-\infty}^{\infty} R_s(\tau) e^{-i\tau\omega} \tag{5.9}$$

By definition of the inverse Fourier transform it holds, that

$$R_s(0) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \mathcal{E}\{s^2(t)\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_s(\omega)\, d\omega \tag{5.10}$$

The cross spectrum between two jointly quasi-stationary signals $s(t)$ and $w(t)$ is similarly defined as

$$\Phi_{sw}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{sw}(\tau) e^{-i\tau\omega} \tag{5.11}$$

$\square$

As it appears, in this chapter we work with frequencies normalized with respect to the sampling frequency $1/T$ where T is the sampling period. Hence, the primary band of discrete-time angular frequencies range from $-\pi$ to π. We have the following important result for the spectrum of a filtered signal.

Theorem 12 *(Ljung 1987)*

Let $s(t) = H(q)v(t)$ where $H(q)$ is a stable transfer function and $v(t)$ is a quasi-stationary signal with spectrum $\Phi_v(\omega)$. Then $s(t)$ is a quasi-stationary signal with spectrum

$$\Phi_s(\omega) = |H(e^{iw})|^2 \Phi_v(\omega) \tag{5.12}$$

$\square$

This result is used extensively in the following.

5.2.3 Prediction error identification methods

Most of the identification methods we consider here are variants of the so-called prediction error method as described in (Ljung 1987). Consider the model description

$$y(t) = \hat{G}(q,\theta)u(t) + \hat{H}(q,\theta)e(t) \tag{5.13}$$

Define the associated one step ahead predictor

$$\hat{y}(t|\theta) = \left(1 - \hat{H}^{-1}(q,\theta)\right) y(t) + \hat{H}^{-1}(q,\theta)\hat{G}(q,\theta)u(t) \tag{5.14}$$

with prediction error

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta) \tag{5.15}$$

The system is identified based on a set of collected data contained in the signal vector Z^N

$$Z^N = [y(T), u(T), y(2T), u(2T), \ldots, y(NT), u(NT)] \tag{5.16}$$

where T is the sampling period. We seek a map from this set of data into the parameter space

$$Z^N \to \hat{\theta}_N \in D \tag{5.17}$$

where D is a set of feasible parameter vectors. Such a mapping is a parameter estimation method.

Define the filtered prediction error

$$\varepsilon_f(t,\theta) = L(q)\varepsilon(t,\theta) \tag{5.18}$$

where $L(q)$ is a stable linear filter. The criterion function for the identification problem is the following scalar-valued function of the model parameters

$$J_N(\theta, Z^N) = \frac{1}{N}\sum_{t=1}^{N} \ell(\varepsilon_f(t,\theta)) \tag{5.19}$$

where $\ell(\cdot)$ is a scalar-valued function. The estimate $\hat{\theta}_N$ is defined by minimization of the above criterion

$$\hat{\theta}_N = \hat{\theta}_N(Z^N) = \arg\min_{\theta \in D} J_N(\theta, Z^N) \tag{5.20}$$

Such approaches are called prediction error methods (PEM). A common choice for the function $\ell(\cdot)$ is the quadratic norm

$$\ell(\varepsilon) = \frac{1}{2}\varepsilon^2 \tag{5.21}$$

resulting in least-squares type identification algorithms.

5.2.4 Bias/variance errors

Denote an identified not necessarily parametric model by $\hat{G}_N(e^{i\omega})$ where N is the size of the data set. Introduce the error of the identified model

$$\tilde{G}_N(e^{i\omega}) = \hat{G}_N(e^{i\omega}) - G_0(e^{i\omega}) \tag{5.22}$$

This error can naturally be split up in a part due to randomness in data and a part due to bias

$$\tilde{G}_N(e^{i\omega}) = \underbrace{\hat{G}_N(e^{i\omega}) - \mathcal{E}\{\hat{G}_N(e^{i\omega})\}}_{\text{random error}} + \underbrace{\mathcal{E}\{\hat{G}_N(e^{i\omega})\} - G_0(e^{i\omega})}_{\text{bias error}} \tag{5.23}$$

Define the bias of the estimate as

$$B_N(\omega) = \mathcal{E}\{\hat{G}_N(e^{i\omega})\} - G_0(e^{i\omega}) \tag{5.24}$$

The term bias is usually defined as a description of parameter errors in a given parametrization of the model. We extend as in (Wahlberg and Ljung 1986) the use of this term to generally describe systematic deviances of the model estimate not attributable to random effects. Define also the variance of the model estimate

$$P_N(\omega) = \mathcal{E}\left\{\left|\hat{G}_N(e^{i\omega}) - \mathcal{E}\{\hat{G}_N(e^{i\omega})\}\right|^2\right\} \tag{5.25}$$

With these definition we find, that the mean square error of the estimate can be expressed as (Ljung 1985)

$$\mathcal{E}\left\{\left|\hat{G}_N(e^{i\omega}) - \hat{G}_0(e^{i\omega})\right|^2\right\} = |B_N(\omega)|^2 + P_N(\omega) \tag{5.26}$$

The analysis of this error has been studied in (Ljung 1985) and (Wahlberg and Ljung 1986). The paper by Ljung focuses on the variance error and derives asymptotic (in both model order and number of observed data) expressions when the true system is contained in the model set. The companion paper by Wahlberg and Ljung is concerned with the bias-part and in particular how the distribution of this error over frequency can be affected by the designer of the experimental conditions. The paper (Gevers and Ljung 1986) uses the asymptotic expression for bias- and variance-errors derived in (Ljung 1985) and (Wahlberg and Ljung 1986) to give guidelines on how to design optimal experiments for transfer function estimation. The use of a model for design in a given application gives rise to a performance degradation compared to using knowledge of the true system. (Gevers and Ljung 1986) shows how to minimize this degradation by proper experiment design. They consider both the bias and the variance contribution. Gevers and Ljung present an interesting numerical example where the intended use of the model is to design a minimum-variance regulator. They show, that the optimal design is to perform identification under optimal minimum-variance control, that is closed-loop identification is superior to open-loop identification. However, since the optimal controller depends on the best model for control, this result encourages iterative approaches.

In the remains of the chapter we will focus on the bias problem. In (Hjalmarsson *et al.* 1994a) some results for the variance aspect are presented. They assume, that the true system is contained in the model set so that bias can be ignored. For minimum-variance control, model reference control and LQG control they find, that the best performance is achieved by closed-loop identification preferably with the ideal controller. The measure of performance is the variance of the error between the ideal closed-loop system (with ideal controller) and the achieved closed-loop system (with the model-based controller).

5.3 Distribution of bias in transfer function estimation

In (Wahlberg and Ljung 1986) a detailed analysis of the distribution of bias over frequency in off-line transfer function estimation is presented. In particular, the effect of prefiltering the data is discussed. The results have later been presented in (Ljung 1987). The observations made by Wahlberg and Ljung, although coming from open-loop considerations, have accelerated the research on control-relevant prefilter design. Later in this chapter several of the proposed approaches are examined. Because of the widespread usage of the results presented in (Wahlberg and Ljung 1986) they are conveyed in some detail below.

Assume, that the true system can be described by

$$y(t) = G_0(q)u(t) + v(t) \tag{5.27}$$

where the disturbance term $v(t)$ can be described by a stochastic process with zero mean and spectrum $\Phi_v(\omega)$. Furthermore, assume that we seek to find the best possible system model within a set of parameterized models described by

$$y(t) = \hat{G}(q,\theta)u(t) + \hat{H}(q,\theta)e(t) \tag{5.28}$$

where $\{e(t)\}$ is a white noise sequence. θ is the parameter vector. For instance, an ARMAX model is described by

$$y(t) = \frac{B(q,\theta)}{A(q,\theta)}u(t) + \frac{C(q,\theta)}{A(q,\theta)}e(t) \tag{5.29}$$

where A, B and C are polynomials in q. The aim of the estimation algorithm is to find a parameter vector such that

$$\hat{\theta}_N = \arg\min_{\theta} J_N(\theta) \tag{5.30}$$

with the identification criterion

$$J_N(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[L(q)\varepsilon_k(t,\theta)\right]^2 = \frac{1}{N}\sum_{i=1}^{N}\left[\varepsilon_k^f(t,\theta)\right]^2 \tag{5.31}$$

where $\varepsilon_k(t,\theta)$ is the error of the k-step prediction of $y(t)$ given the model parameterized by θ

$$\varepsilon_k(t,\theta) = y(t) - \hat{y}(t|t-kT,\theta) \tag{5.32}$$

where

$$\hat{y}(t|t-kT,\theta) \triangleq \mathcal{E}\{y(t)|\mathcal{I}_{c,t,k}\} \tag{5.33}$$

The information set on which prediction is based is defined as in Section 3.1

$$\mathcal{I}_{c,t,k} = \{y(t), y(t-T), \ldots, u(t+(k-1)T), u(t+(k-2)T), \ldots\} \tag{5.34}$$

where we have assumed a time-delay of one sample of the model. $L(q)$ is a stable, linear prefilter. In other words, we wish to minimize the sum of squared, filtered k-step prediction errors. Note, that with $k = 1$ we recover the usual one-step prediction error method and with $k = \infty$ we get an output-error method.

If the true transfer function is not contained in the model set, then it is clear that there will be some discrepancy between the estimated model and the true system. As defined earlier this discrepancy is called bias. In practice, the estimates will always be biased due to unmodelled dynamics.

As discussed in (Wahlberg and Ljung 1986) the k-step prediction error can be expressed as

$$\varepsilon_k(t,\theta) = \left[\left(G_0(q) - \hat{G}(q,\theta)\right) u(t) + v(t)\right] \frac{\bar{H}_k(q,\theta)}{\hat{H}(q,\theta)} \tag{5.35}$$

where

$$\hat{H}(q,\theta) = \bar{H}_k(q,\theta) + q^{-k}\tilde{H}_k(q,\theta) \tag{5.36}$$

Without loss of generality it can be assumed, that $\bar{H}_k(q,\theta) = 1$. Define the spectrum of the filtered k-step prediction error as $\Phi_\varepsilon^f(\omega,\theta)$. Then from the above it appears, that

$$\Phi_\varepsilon^f(\omega,\theta) = \left[|G_0(e^{i\omega}) - \hat{G}(e^{i\omega},\theta)|^2 \Phi_u(\omega) + \Phi_v(\omega)\right] |L(e^{i\omega})|^2 \left|\frac{\bar{H}_k(e^{i\omega},\theta)}{\hat{H}(e^{i\omega}\theta)}\right|^2 \tag{5.37}$$

where $\Phi_u(\omega)$ is the spectrum of the input signal.

Distribution of bias over frequency

The following result is proved in (Wahlberg and Ljung 1986)

Theorem 13 *(Wahlberg and Ljung 1986)*

If the input signal $u(t)$ is independent from $\{v(t)\}$ then $\hat{\theta}_N \to \theta^*$ with probability 1, as $N \to \infty$ where

$$\theta^* = \arg\min_{\theta} \bar{J}(\theta) \tag{5.38}$$

with

$$\begin{aligned} \bar{J}(\theta) &= \int_{-\pi}^{\pi} \Phi_{\varepsilon}^{f}(\omega)\, d\omega \\ &= \int_{-\pi}^{\pi} \left[|G_0 - \hat{G}(\theta)|^2 \Phi_u + \Phi_v\right] |L|^2 \left|\frac{\bar{H}_k(\theta)}{\hat{H}(\theta)}\right|^2 d\omega \end{aligned} \tag{5.39}$$

□

Here dependencies on ω has been dropped for convenience. For the one-step-ahead least squares estimator ($k = 1$) we get the following asymptotic loss function

$$\bar{J}(\theta) = \int_{-\pi}^{\pi} \left[|G_0 - \hat{G}(\theta)|^2 \Phi_u + \Phi_v\right] \left|\frac{L}{\hat{H}(\theta)}\right|^2 d\omega \tag{5.40}$$

To analyze (5.39) it is convenient to look at two special cases.

Fixed noise model

Suppose the noise model $\hat{H}(q, \theta)$ is fixed to $\hat{H}(q, \theta_F)$. Then it is apparent from (5.39), that asymptotically the difference between the true transfer function $G_0(e^{i\omega})$ and the limiting model $\hat{G}(e^{i\omega}, \theta)$ is minimized in a quadratic norm with weighting function

$$Q(\omega, \theta_F) = \Phi_u |L|^2 \left| \frac{\bar{H}_k(\theta_F)}{\hat{H}(\theta_F)} \right|^2 \tag{5.41}$$

Under these circumstances the noise spectrum $\Phi_v(\omega)$ does not influence the minimization. Furthermore, it appears that prefiltering is equivalent to changing the noise model.

Suppose now, that $L(q) \equiv 1$ and the input is white noise (i.e. $\Phi_u(\omega)$ is constant). Then if $k = 1$ (least squares method) the weighting function Q is proportional to

$$Q(\omega, \theta_F) \propto \frac{1}{\hat{H}(e^{i\omega}, \theta_F)} \tag{5.42}$$

With a physical low-pass noise model we see, that the weighting function causes an emphasis on high-frequencies. This is a common observation for one-step-ahead prediction error methods, see e.g. (Mohtadi 1988).

If instead $k \to \infty$ (output error method), then

$$Q(\omega, \theta_F) \propto 1 \tag{5.43}$$

and all frequencies will be equally weighted.

In the area in between, $1 < k < \infty$, we will get a weighting function in between the two above weighting functions. We conclude, that using a k-step prediction error method moves emphasis from high frequencies to low frequencies when k is increased.

Fixed deterministic transfer function

If the transfer function $\hat{G}(q, \theta)$ is fixed to $\hat{G}(q, \theta_F)$ and the noise transfer function is variable, then the analysis is somewhat more complicated. Essentially, the spectrum of the noise transfer function is fitted to the spectrum of the filtered error signal

$$L(q)\left(y(t)-\hat{G}(q,\theta_F)u(t)\right) \tag{5.44}$$

with spectrum

$$\Phi_{er}(\omega)=\left[|G_0-\hat{G}(\theta_F)|^2\Phi_u+\Phi_v\right]|L|^2 \tag{5.45}$$

For further details, see (Wahlberg and Ljung 1986).

General minimization problem

In the general case, where both the deterministic and the noise transfer function depend on the parameter vector θ it is not possible to give a simple interpretation of how θ^* is chosen. However, the two above mentioned special cases show, that the minimization problem (5.38) can be seen as a compromise between fitting the deterministic transfer function $\hat{G}(q,\theta)$ to the true transfer function $G_0(q)$ in a frequency weighted norm and of fitting the noise spectrum $|\hat{H}(e^{i\omega},\theta)|^2$ to the spectrum of the error signal given in (5.44). The weighting function $Q(e^{i\omega},\theta^*)$

$$Q(\omega,\theta^*)=\Phi_u|L|^2\left|\frac{\bar{H}_k(\theta^*)}{\hat{H}(\theta^*)}\right|^2 \tag{5.46}$$

is partly or entirely determining for the distribution of bias over frequencies.

It should be mentioned, that the results given here are valid for the off-line case. In the on-line case it is practically impossible at each time instant to calculate the filtered prediction error based on the most recent parameter estimate. This will usually (except for finite impulse response filters) involve processing all gathered data. Instead, old prediction errors based on older parameter estimates are used in the filtering operation. However, the difference between the on-line and the off-line case will vanish as $N\to\infty$ if the parameter estimates converge. Thus, the off-line analysis is also useful when considering on-line estimation.

Bode-plot interpretation

It is often of more interest to consider

$$\log \hat{G}(e^{i\omega}) = \log |\hat{G}(e^{i\omega})| + i \arg \hat{G}(e^{i\omega}) \tag{5.47}$$

as a function of $\log \omega$ in a Bode plot instead of directly considering $\hat{G}(e^{i\omega})$. In (Wahlberg and Ljung 1986) it has been shown, that a quadratic fit between the amplitudes of the true and estimated transfer function in a Bode plot gives rise to a weighting function

$$\bar{Q}(\log \omega) = Q(\omega)\omega |G_0(e^{i\omega})|^2 \tag{5.48}$$

Consequently, if we want to minimize the relative error and compensate for a logarithmic frequency scale then the weighting function of interest is $\bar{Q}(\log \omega)$. We see, that equal weight over the frequency range in a Bode plot (i.e. for instance the intervals [0.1, 1] and [10, 100] are weighted equally) corresponds to a weighting $Q(\omega)$ which is large at low frequencies and at frequencies where the true transfer function is small.

Choice of prefilter

It has appeared, as expected, that the prefilter $L(q)$ plays an important role in estimating transfer functions in the face of unmodelled dynamics. The prefilter should be chosen in accordance with the application of the estimated model to give a sensible distribution of bias.

In control systems an important use of the prefilter is to mitigate the effects of e.g. unmodelled dynamics and disturbances. Most disturbances appear at either very low frequencies or at high frequencies, (Mohtadi 1988). Unmodelled components will usually be located at high frequencies. An *ad hoc* guideline for the choice of prefilter is therefore a low-pass or band-pass filter with an upper break frequency around the closed-loop bandwidth.

Later in this chapter we will discuss how to select the data filter in accordance with the control objective. The first thing to discuss is how the bias-distribution is affected by the prefilter when the system is operating in closed-loop.

5.4 Closed-loop identification

As mentioned the results in Section 5.3 require independence between the input $u(t)$ and the noise sequence $\{v(t)\}$, i.e. are only valid in the open-loop case. Similar results can be obtained in the closed-loop case. Here we will limit ourselves to the one-step-ahead prediction case, i.e. $k = 1$. Assume, that we have a fixed controller

$$u(t) = -C_2 y(t) + C_1 r(t) \tag{5.49}$$

where C_2 and C_1 are transfer functions in the shift operator and $r(t)$ is the reference, which is assumed to be independent from $\{v(t)\}$. We then get the achieved closed-loop system depicted in Figure 5.2.

With the real system given by (5.27) we get

$$y(t) = \frac{1}{1+G_0C_2}\left(G_0C_1 r(t) + v(t)\right) \tag{5.50}$$

By using this equation and (5.27) we can express the filtered one-step prediction as[1]

$$\begin{aligned}
\varepsilon_f(t) &= \left[\left(G_0 - \hat{G}\right) u(t) + v(t)\right] \frac{L}{\hat{H}} \\
&= \left[\left(G_0 - \hat{G}\right)\left(\frac{-C_2}{1+G_0C_2}\left(G_0C_1 r(t) + v(t)\right) + C_1 r(t)\right) + v(t)\right] \frac{L}{\hat{H}} \\
&= \left[\left(G_0 - \hat{G}\right)\frac{C_1}{1+G_0C_2} r(t) + \left(1 - \frac{(G_0 - \hat{G})C_2}{1+G_0C_2}\right) v(t)\right] \frac{L}{\hat{H}}
\end{aligned}$$

[1]Here and in the following the dependence on θ in $\hat{G}(q,\theta)$ is left out where convenient.

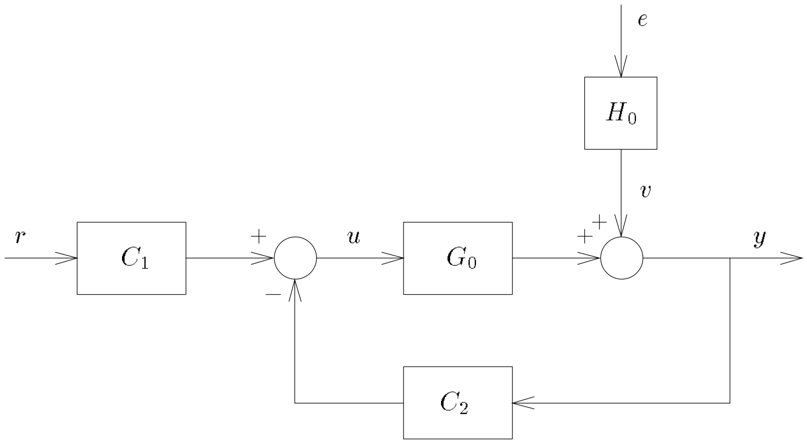

Figure 5.2: Achieved closed-loop system.

$$= \frac{1}{1+G_0C_2}\left[\left(G_0-\hat{G}\right)C_1r(t)+\left(1+\hat{G}C_2\right)v(t)\right]\frac{L}{\hat{H}} \tag{5.51}$$

The spectrum of this prediction error is

$$\Phi_\varepsilon^f = \left|\frac{L}{\hat{H}}\right|^2\left[|\hat{G}-G_0|^2\left|\frac{C_1}{1+G_0C_2}\right|^2\Phi_r+\left|\frac{1+\hat{G}C_2}{1+G_0C_2}\right|^2\Phi_v\right] \tag{5.52}$$

The asymptotic parameter estimate is then given by

$$\theta^* = \arg\min_\theta \int_{-\pi}^{\pi}\Phi_\varepsilon^f(\omega,\theta)\,d\omega = \arg\min_\theta \bar{J}_{cl}(\theta) \tag{5.53}$$

where

$$\bar{J}_{cl} = \int_{-\pi}^{\pi}\left|\frac{1}{1+G_0C_2}\right|^2\left[|G_0-\hat{G}|^2|C_1|^2\Phi_r+\left|1+\hat{G}C_2\right|^2\Phi_v\right]\left|\frac{L}{\hat{H}}\right|^2 d\omega \tag{5.54}$$

As it appears this expression is quite complex, and it is in general difficult to analyze the effect of the controller in the estimation problem. Some special cases can be discussed.

If $\Phi_v(\omega) \gg \Phi_r(\omega)$ then

$$\bar{J}_{cl} \approx \int_{-\pi}^{\pi} \left| \frac{1+\hat{G}C_2}{1+G_0C_2} \right|^2 \Phi_v \left| \frac{L}{\hat{H}} \right|^2 d\omega \tag{5.55}$$

Minimization of this criterion yields

$$\hat{G} \approx -\frac{1}{C_2} \tag{5.56}$$

where the weight is high, i.e. we identify the inverse controller. This is the well-known identifiability problem, i.e., not surprisingly, in frequency ranges where the signal to noise ratio is low it is difficult to extract information about the system.

If $\Phi_r(\omega) \gg \Phi_v(\omega)$ then

$$\bar{J}_{cl} \approx \int_{-\pi}^{\pi} \left| \frac{1}{1+G_0C_2} \right|^2 |G_0 - \hat{G}|^2 |C_1|^2 \Phi_r \left| \frac{L}{\hat{H}} \right|^2 d\omega \tag{5.57}$$

In this case, the difference between the identified model and the true system will be minimized in a quadratic norm weighted by $|L/\hat{H}|$, the spectrum of the filtered reference $C_1 r(t)$ and by $1/|1+G_0C_2|$. This last term is the closed-loop sensitivity function. The reference tracking and disturbance rejection is good where the sensitivity function is small. Hence, this weighting implies, that the estimation tries to minimize the error between G_0 and $\hat{G}$ where the control is poor, which potentially may improve the control performance. This supports the by now common observation, that identification should take place in closed-loop if the intended use of the model is control design.

It is clear that the reference signal plays an important role and should excite the system in the relevant frequency areas. In control systems this means within the closed-loop bandwidth. In lack of excitation in the reference signal it may be better to shut off the estimation algorithm or to inject a properly chosen external perturbation signal.

Consider the following alternative expression for the filtered one-step prediction error (5.51)

$$\begin{aligned}\varepsilon_f(t) &= \frac{L}{\hat{H}(1+G_0C_2)}\Big[\left(G_0-\hat{G}\right)C_1r(t)\\ &\qquad + \left(H_0(1+\hat{G}C_2)-\hat{H}(1+\hat{G}C_2)\right)e(t)\Big] + Le(t) \end{aligned} \quad (5.58)$$

where we have used $v(t) = H_0(q)e(t)$ with $e(t)$ being a white noise signal with variance σ_e^2. The last term is independent of θ and hence as noted in (Gevers 1993) minimizing $\bar{J}_{cl}$ is equivalent to minimizing

$$\begin{aligned}\bar{J}_{cl}' &= \int_{-\pi}^{\pi}\left\{\left|\frac{(G_0-\hat{G})}{1+G_0C_2}\right|^2|C_1|^2\Phi_r\right.\\ &\qquad \left. + \left|\frac{H_0(1+\hat{G}C_2)-\hat{H}(1+G_0C_2)}{1+G_0C_2}\right|^2\sigma_e^2\right\}\left|\frac{L}{\hat{H}}\right|^2 d\omega \end{aligned} \quad (5.59)$$

This expression shows, that even if G_0 is contained in the model set then closed-loop identification using a direct prediction error method will lead to a biased estimate if $\hat{H} \neq H_0$, that is if the noise model is incorrect. An alternative indirect estimation approach which alleviates this problem is discussed in the next section. Other approaches which transforms the closed-loop estimation problem into an open-loop one by coprime factorization or Youla parametrizations have also been suggested, see (Van den Hof and Schrama 1995). This has been used in the so-called Schrama scheme for iterative identification and control, see Section 5.7.2.

5.5 Indirect closed-loop identification

Another approach to closed-loop identification is taken in (Van den Hof and Schrama 1993). Here a two-stage indirect approach to closed-loop estimation is developed. It will appear, that this method has certain advantages as compared to the direct approach described in the previous section.

The first step in this approach is to identify the closed-loop sensitivity function. Consider the true system description

$$y(t) = G_0(q)u(t) + H_0(q)e(t) \tag{5.60}$$

where $e(t)$ is a white noise signal. The control signal is generated as the one-degree of freedom control law

$$u(t) = r(t) - C(q)y(t) \tag{5.61}$$

The closed-loop description becomes

$$y(t) = \frac{G_0(q)}{1 + G_0(q)C(q)}r(t) + \frac{H_0(q)}{1 + G_0(q)C(q)}e(t) \tag{5.62}$$

for the output and

$$u(t) = \frac{1}{1 + G_0(q)C(q)}r(t) - \frac{H_0(q)C(q)}{1 + G_0(q)C(q)}e(t) \tag{5.63}$$

for the input. Introduce the closed-loop sensitivity function

$$S_0(q) = \frac{1}{1 + G_0(q)C(q)} \tag{5.64}$$

Now,

$$u(t) = S_0(q)r(t) - H_0(q)C(q)S_0(q)e(t) \tag{5.65}$$

From this equation it appears, that $S_0(q)$ can be estimated consistently using a model structure with independently parameterized transfer function from $r(t)$ and $e(t)$ (Ljung 1987)

$$u(t) = S(q, \beta)r(t) + R(q, \gamma)\varepsilon_u(t) \tag{5.66}$$

with β and γ being the parameter vectors. The estimate $\hat{S}(q, \beta)$ is determined as a least-squares fit according to

$$\hat{\beta}_N = \arg_\beta \min_{\beta,\gamma} \frac{1}{N} \sum_{t=1}^{N} \varepsilon_u^2(t) \tag{5.67}$$

In (Van den Hof and Schrama 1993) it is suggested to use high-order FIR filters, possibly with a shift to for instance Laguerre basis functions, in the estimation of the sensitivity function $S_0(q)$.

We now separate the effect of the excitation $r(t)$ and the disturbance $e(t)$ on the plant input. An estimate of the disturbance-free plant input is given by

$$\hat{u}_N^r(t) = S(q, \hat{\beta}_N)r(t) \tag{5.68}$$

Consider now the model structure

$$y(t) = \hat{G}(q, \theta)\hat{u}_N^r(t) + \hat{H}(q, \eta)\varepsilon_y(t) \tag{5.69}$$

where $\hat{G}(q, \theta)$ and $\hat{H}(q, \eta)$ are parameterized independently.

In the second step of the indirect identification procedure a plant estimate is determined as

$$\hat{\theta}_N = \arg_\theta \min_{\theta,\eta} \frac{1}{N} \sum_{t=1}^{N} \varepsilon_y^2(t) \tag{5.70}$$

The following consistency result can be proved.

Theorem 14 *(Van den Hof and Schrama 1993)*

Given that $S_0(q)$ is asymptotically stable, $e(t)$ and $r(t)$ are uncorrelated quasi-stationary signals and $r(t)$ is persistently exciting. Then if S_0 and G_0 are covered by the model structures (5.66) and (5.69) it holds under weak conditions, that

$$\lim_{N\to\infty} \hat{G}(q, \hat{\theta}_N) = G_0(q) \qquad \text{w.p. } 1 \tag{5.71}$$

□

Such a consistency result can not be obtained in the direct closed-loop prediction error identification approach as pointed out by (5.59).

Considering the case of undermodelling in both steps of the identification procedure gives rise to the following result for the frequency characterization of bias in the estimation problem

Theorem 15 *(Van den Hof and Schrama 1993)*

If fixed noise models is used in both estimation steps, i.e. $R(q,\gamma) = K(q)$ *and* $\hat{H}(q,\eta) = D(q)$ *then, under weak conditions,* $\hat{\theta}_N \to \theta^*$ *with probability 1 as* $N \to \infty$ *with*

$$\theta^* = \arg\min_{\theta} \int_{-\pi}^{\pi} \left| G_0(e^{i\omega})S_0(e^{i\omega}) - \hat{G}(e^{i\omega},\theta)S(e^{i\omega},\beta^*) \right|^2 \frac{\Phi_r(\omega)}{|D(e^{i\omega})|^2}\, d\omega \tag{5.72}$$

and

$$\beta^* = \arg\min_{\beta} \int_{-\pi}^{\pi} \left| S_0(e^{i\omega}) - S(e^{i\omega},\beta) \right|^2 \frac{\Phi_r(\omega)}{|K(e^{i\omega})|^2}\, d\omega \tag{5.73}$$

□

It appears, that the bias distribution of $\hat{G}(q,\hat{\theta})$ is dependent on the accuracy of the estimated sensitivity function from the first identification step. Also note, that the bias distribution is independent of the spectrum of the disturbance signal $\Phi_v(\omega) = |H_0(e^{i\omega})|^2\Phi_e(\omega)$. This is not the case for the conventional direct estimation of $G_0(q)$, see (5.54).

If the first step of the identification procedure is done without error, that is $T(q,\beta^*) = T_0(q)$, then the frequency characterization of the bias distribution reduces to

$$\theta^* = \arg\min_{\theta} \int_{-\pi}^{\pi} \left| G_0(e^{i\omega}) - \hat{G}(e^{i\omega},\theta) \right|^2 \frac{|S_o(e^{i\omega})|^2\Phi_r(\omega)}{|D(e^{i\omega})|^2}\, d\omega \tag{5.74}$$

We see, that in this situation the error of the transfer-function estimate $\hat{G}(e^{i\omega},\theta)$ is frequency weighted by the true closed-loop sensitivity function. Furthermore, the filter $D(q)$ can be used directly to shape the distribution of the error.

5.6 Control-relevant prefiltering

The asymptotic analysis of the frequency characterization of the bias-distribution in transfer function estimation introduced by (Wahlberg and Ljung 1986) has had a large impact on the area of identification for control purposes. In particular the description of the effect of prefiltering data gives us a tool for shaping bias-distribution to obtain more appropriate models for control design. Several angles of attack have been proposed in the literature of which some are described in this section. Others not treated here include (Rivera *et al.* 1992, Vodă and Landau 1995, Van der Klauw *et al.* 1994, Bitmead *et al.* 1990). The latter presents some general guidelines for choice of prefilter for the CRAP (Candidate Robust Adaptive Predictive) controller, mainly from robust stability considerations. Also the so-called Zang-scheme for iterative identification and control includes control-relevant prefiltering. This algorithm is discussed in detail in Section 5.7.3.

5.6.1 An optimal choice of prefilter with fixed controller

In this section we present a way to choose the prefilter in closed-loop estimation with a fixed controller. The choice is based on a performance requirement, namely the minimization of a closed-loop output error. This result may be considered a generalization of the results presented by (Åström 1993), see Section 5.6.2, with a frequency domain interpretation.

Consider the output-error system

$$y(t) = G_0(q)u(t) + v(t) \tag{5.75}$$

where $v(t)$ is a noise signal with spectrum $\Phi_v(\omega)$. With the fixed controller

$$u(t) = C_1(q)r(t) - C_2(q)y(t) \tag{5.76}$$

we get the achieved closed-loop equation

$$y(t) = R(q)r(t) + S(q)v(t) \tag{5.77}$$

with

$$R(q) = \frac{G_0(q)C_1(q)}{1 + G_0(q)C_2(q)} \tag{5.78}$$

$$S(q) = \frac{1}{1 + G_0(q)C_2(q)} \tag{5.79}$$

Using the deterministic model

$$y_c(t) = \hat{G}(q)u_c(t) \tag{5.80}$$

and corresponding controller

$$u_c(t) = C_1(q)r(t) - C_2(q)y_c(t) \tag{5.81}$$

the nominal closed-loop equation becomes

$$y_c(t) = R_c(t)r(t) \tag{5.82}$$

with

$$R_c(t) = \frac{\hat{G}(q)C_1(q)}{1 + \hat{G}(q)C_2(q)} \tag{5.83}$$

We now set up the error signal $\tilde{y}(t)$ between the nominal closed-loop output and the obtained closed-loop output, see Figure 5.3,

$$\begin{aligned}
\tilde{y}(t) &= y_c(t) - y(t) \\
&= \left(\frac{\hat{G}C_1}{1 + \hat{G}C_2} - \frac{G_0C_1}{1 + G_0C_2} \right) r(t) - \frac{1}{1 + G_0C_2} v(t) \\
&= \frac{C_1\hat{G}(1 + G_0C_2) - G_0C_1(1 + \hat{G}C_2)}{(1 + \hat{G}C_2)(1 + G_0C_2)} r(t) - \frac{1}{1 + G_0C_2} v(t) \\
&= \frac{(\hat{G} - G_0)C_1}{(1 + \hat{G}C_2)(1 + G_0C_2)} r(t) - \frac{1}{1 + G_0C_2} v(t)
\end{aligned} \tag{5.84}$$

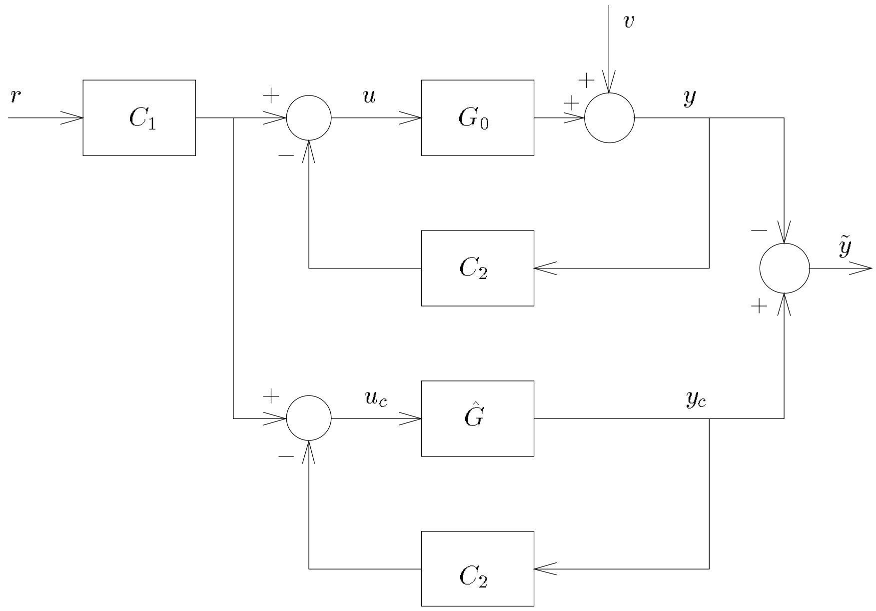

Figure 5.3: Generation of the error between the nominal and the obtained closed-loop output.

Using Parsevals Theorem we get the following spectrum of the error signal $\tilde{y}(t)$

$$\Phi_{\tilde{y}} = \frac{|\hat{G} - G_0|^2 |C_1|^2}{|1 + \hat{G}C_2|^2 |1 + G_0 C_2|^2} \Phi_r + \frac{1}{|1 + G_0 C_2|^2} \Phi_v \tag{5.85}$$

where we have assumed that $r(t)$ and $v(t)$ are uncorrelated. A performance requirement for an optimal choice of prefilter is to choose L in (5.52) such that the variance of $\tilde{y}(t)$ is minimized. Now, this variance is given by

$$\mathrm{V}\{\tilde{y}(t)\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{\tilde{y}}(\omega)\, d\omega \tag{5.86}$$

Comparison to (5.52) shows, that with the prefilter chosen as

$$\boxed{\left| \frac{L}{\hat{H}} \right|^2 = \frac{1}{|1 + \hat{G}C_2|^2}} \tag{5.87}$$

we get $\Phi_\varepsilon^f = \Phi_{\tilde{y}}$. Hence, according to (5.53), with this choice of prefilter the variance of $\tilde{y}(t)$ is minimized by the one-step prediction error estimation algorithm. For instance, with $\hat{H} = 1$, that is output error identification, the prefilter should be chosen as the *nominal* closed-loop sensitivity function. Usually, for physical systems the noise model will have low-pass characteristics. The sensitivity function is normally of high-pass character with a high-frequency amplification of 1. Hence, often the above of prefilter will be of band-pass type. Using band-pass data-filters is commonly advocated in ad hoc prefilter design as such filters attenuate disturbances where these often appear: at low frequencies (e.g. load-disturbances or offsets in measurement) and at high frequencies (e.g. electrical noise or quantization noise), see (Mohtadi 1988).

5.6.2 Matching criteria for identification and control

(Åström 1993) presents a time-domain analysis for finding control-relevant prefilters in estimation with a pole-placement control design. By comparing the error signal minimized in mean-square by the estimation algorithm with the control-performance error he derives criteria for matching identification and control.

Consider the plant model

$$y(t) = \frac{\hat{B}(q)}{\hat{A}(q)}u(t) + \frac{C(q)}{\hat{A}(q)}e(t) \tag{5.88}$$

where $e(t)$ is white noise. We assume, that $C(q)$ is known. The filtered prediction error minimized in mean-square by the estimation algorithm becomes

$$e_{fp} = L(q)\left(\frac{\hat{A}}{C}y(t) - \frac{\hat{B}}{C}u(t)\right) \tag{5.89}$$

where $L(q)$ is the transfer function of the data filter. We design a pole-placement controller

$$R(q)u(t) = Q(q)r(t) - S(q)y(t) \tag{5.90}$$

where R, Q and S are design polynomials. The closed-loop characteristic polynomial is

$$A_c = \hat{A}R + \hat{B}S \tag{5.91}$$

We choose the observer polynomial $Q = C$ and specify $A_c = CA_m$ where A_m contains the desired closed-loop poles. Let $\hat{G} = \hat{B}/\hat{A}$. In the following assume for simplicity that $e(t) = 0$. The model output in closed-loop becomes

$$y(t) = \frac{\hat{G}C}{R + \hat{G}S}r(t) \tag{5.92}$$

By comparing the responses of the designed and achieved closed-loop we get the control performance error

$$e_{cp} = y_0(t) - y(t) \tag{5.93}$$

where y_0 is the response of the true plant. Denote the true plant by $G_0 = B/A$. We then have

$$y_0(t) = \frac{G_0 C}{R + G_0 S} r(t) \tag{5.94}$$

$$u_0(t) = \frac{C}{R + G_0 S} r(t) \tag{5.95}$$

and hence

$$\begin{aligned} e_{cp} &= \left(\frac{G_0 C}{R + G_0 S} - \frac{\hat{G} C}{R + \hat{G} S} \right) r(t) \\ &= \frac{RC(G_0 - \hat{G})}{(R + G_0 S)(R + \hat{G} S)} r(t) \\ &= \frac{RC(\hat{A} G_0 - \hat{B})}{A_m C(R + G_0 S)} r(t) \\ &= \frac{R}{A_m} \left(\frac{\hat{A}}{C} y_0(t) - \frac{\hat{B}}{C} u_0(t) \right) \end{aligned} \tag{5.96}$$

On comparing the filtered prediction error (5.89) with this control performance error it appears, that they match if the filter L is chosen as

$$\boxed{L = \frac{R}{A_m}} \tag{5.97}$$

Possible convergence points of an iterative procedure with this prefilter have been analysed in (Åström and Nilsson 1994) for a second order system with a first order model. They arrive at upper bound on the designed bandwidth for which possible fix-points results in stable models. The results of (Åström 1993) have been used in (Vodă and Landau 1994) where some application examples are shown.

The optimal choice of prefilter for estimation presented in the previous section can actually be viewed as a slightly more general frequency domain interpretation of Åströms observations. With $\hat{G} = \hat{B}/\hat{A}$, $\hat{H} = C/\hat{A}$, $C_1 = C/R$, $C_2 = S/R$ and a poleplacement design with $A_c = A_m C = \hat{A}R + \hat{B}S$ we get according to (5.87)

$$L = \frac{\hat{H}}{1 + \hat{G}C_2} = \frac{\frac{C}{\hat{A}}}{1 + \frac{\hat{B}S}{\hat{A}R}} = \frac{CR}{\hat{A}R + \hat{B}S} = \frac{CR}{A_c} = \frac{R}{A_m} \tag{5.98}$$

which is exactly the prefilter found by Åström. Note, that the filter by construction is stable as A_c is the designed closed-loop characteristic polynomial.

5.6.3 Approximate identification with a closed-loop performance criterion

Another approach to control-relevant prefiltering was presented in (Hakvoort *et al.* 1992, Hakvoort *et al.* 1994). They set up a closed-loop performance criterion for the identification problem and find the operational conditions and prefiltering choice under which this criterion is minimized.

Consider the closed-loop system depicted in Figure 5.4. The system is working under some operational conditions with noise input $\bar{e}(t)$ and external reference input $\bar{r}(t)$. The spectra of these signals are fixed and entirely determined by the operational conditions of the plant. The achieved closed-loop system equation becomes

$$\bar{y}(t) = \frac{G_0(q)C_1(q)}{1 + G_0(q)C_2(q)}\bar{r}(t) + \frac{H_0(q)}{1 + G_0(q)C_2(q)}\bar{e}(t) \tag{5.99}$$

and the nominal closed-loop equation

$$\bar{y}(t) = \frac{\hat{G}(q,\theta)C_1(q)}{1 + \hat{G}(q,\theta)C_2(q)}\bar{r}(t) + \frac{\hat{H}(q)}{1 + \hat{G}(q,\theta)C_2(q)}\bar{e}(t) \tag{5.100}$$

with fixed noise transfer function $\hat{H}(q)$. The model is a good closed-loop description of the system if the error between the closed-loop deterministic

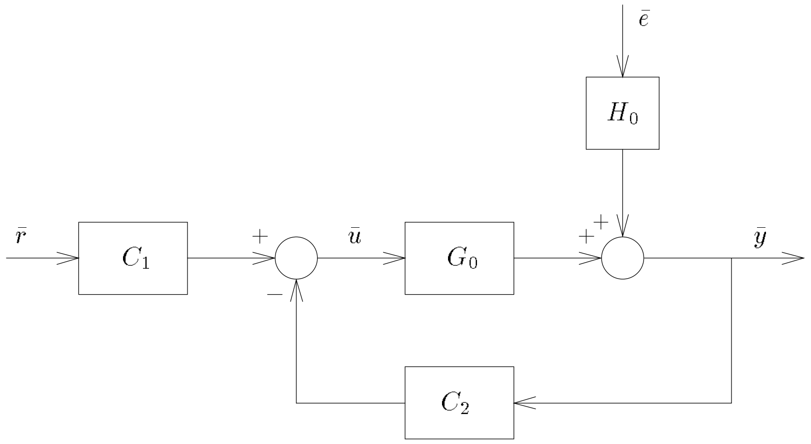

Figure 5.4: Closed-loop system.

transfer functions

$$\frac{\hat{G}(q,\theta)C_1(q)}{1+\hat{G}(q,\theta)C_2(q)} - \frac{G_0(q)C_1(q)}{1+G_0(q)C_2(q)} \tag{5.101}$$

and the error between the closed-loop noise transfer functions

$$\frac{\hat{H}(q)}{1+\hat{G}(q,\theta)C_2(q)} - \frac{H_0(q)}{1+G_0(q)C_2(q)} \tag{5.102}$$

are small. This leads to the model performance criterion (Hakvoort *et al.* 1994)

$$\begin{aligned} J_1(\mathcal{D}) \quad = \quad & \int_{-\pi}^{\pi} \left(\left| \frac{\hat{G}(e^{i\omega},\theta)C_1(e^{i\omega})}{1+\hat{G}(e^{i\omega},\theta)C_2(e^{i\omega})} - \frac{G_0(e^{i\omega})C_1(e^{i\omega})}{1+G_0(e^{i\omega})C_2(e^{i\omega})} \right|^2 \Phi_{\bar{r}}(\omega) \right. \\ & \left. + \left| \frac{\hat{H}(e^{i\omega})}{1+\hat{G}(e^{i\omega},\theta)C_2(e^{i\omega})} - \frac{H_0(e^{i\omega})}{1+G_0(e^{i\omega})C_2(e^{i\omega})} \right|^2 \Phi_{\bar{e}}(\omega) \right) \, d\omega \end{aligned} \tag{5.103}$$

The design variables in consideration for the identification experiment is the input spectrum $\Phi_u(\omega)$ and the cross-spectrum $\Phi_{ue}(\omega)$ due to feedback.

Collect these design-variables in the set

$$\mathcal{D} = \{\Phi_u(\omega),\ \Phi_{ue}(\omega)\} \tag{5.104}$$

Note, that these spectra can be influenced in the identification stage by proper prefiltering (choice of noise-model). This is in contrast with the spectra $\Phi_{\bar{u}}(\omega)$ and $\Phi_{\bar{u}\bar{e}}(\omega)$ which are fixed by the experimental conditions. The objective of the identification problem is to minimize this performance criterion by selecting the optimal design variables, i.e. finding

$$\mathcal{D}_{1,\mathrm{opt}} = \arg\min_{\mathcal{D}} J_1(\mathcal{D}) \tag{5.105}$$

In (Hakvoort *et al.* 1994) it is shown, that asymptotically this performance criterion is minimized by using prediction error identification in closed-loop under normal operating conditions with the prefilter

$$\boxed{L(q) = \frac{\hat{H}(q)}{1+\hat{G}_f(q)C_2(q)}} \tag{5.106}$$

where $\hat{G}_f(q)$ is some fixed model ideally chosen equal to the model identified under optimal operating conditions, $\mathcal{D}_{1,\mathrm{opt}}$. For instance, with output error identification ($\hat{H}(q) = 1$) the optimal filter with respect to minimizing (5.103) is the optimal nominal sensitivity function. Of course, the optimally identified model depends on the filter and noise model used in the identification. This suggests an iterative identification scheme, where $\hat{G}_f(q)$ is updated to the previously identified model at the start of each iteration.

More precisely (Hakvoort *et al.* 1994) finds the following result. Define the discrepancies

$$\delta_1(\omega) = \frac{1}{|1+\hat{G}_f(e^{i\omega})C_2(e^{i\omega})|^2} - \frac{1}{|1+\hat{G}(e^{i\omega},\theta(\mathcal{D}_{1,\mathrm{opt}}))C_2(e^{i\omega})|^2} \tag{5.107}$$

and

$$\delta_2(\omega) = \frac{1}{|1+\hat{G}(e^{i\omega},\theta(\mathcal{D}_{2,\mathrm{opt}}))C_2(e^{i\omega})|^2} - \frac{1}{|1+\hat{G}_f(e^{i\omega})C_2(e^{i\omega})|^2} \tag{5.108}$$

Then the following can be proved.

Theorem 16 *(Hakvoort* et al. *1994)*

Consider the performance criterion $J_1(\mathcal{D})$ as given in (5.103). If the prefilter (5.106) is used in prediction error identification and if the number of samples tends to infinity then the design variables

$$\mathcal{D}_{2,\text{opt}} = \begin{cases} \Phi_{u,\text{opt}}(\omega) = \Phi_{\bar{u}}(\omega) \\ \Phi_{ue,\text{opt}}(\omega) = \Phi_{\bar{u}\bar{e}}(\omega) \end{cases} \tag{5.109}$$

converges to the optimal solution $\mathcal{D}_{1,\text{opt}}$ if $\delta_1(\omega)$ and $\delta_2(\omega)$ converge to zero, i.e.

$$\lim_{\delta_1,\delta_2 \to 0} J_1(\mathcal{D}_{2,\text{opt}}) - J_1(\mathcal{D}_{1,\text{opt}}) = 0 \tag{5.110}$$

□

It is argued in (Hakvoort *et al.* 1994), that the discrepancy $\delta_2(\omega)$ can be reduced to an arbitrarily small value by an iterative procedure in which an identification experiment is repeated based on the same set of closed-loop data. In each iteration the new filter $L(q) = \hat{H}(q)/(1 + \hat{G}_f(q)C_2(q))$ is constructed by the plant estimate of the previous iteration. Note, that δ_2 is computabel, as the design choice $\mathcal{D}_{2,\text{opt}}$ is given by the operational conditions.

The discrepancy $\delta_1(\omega)$ is unknown and depends on the modelling error. Hence, it can generally not be brought to zero. However, if it is small (which happens if both $\hat{G}_f(q)$ and $\hat{G}(q, \theta(\mathcal{D}_{1,\text{opt}}))$ are close to the true system), then the design choice $\mathcal{D}_{2,\text{opt}}$ is expected to be a good choice.

The implication of the result given in Theorem 16 is that the optimal identification strategy is obtained by choosing the input spectrum and the feedback induced cross-spectrum of noise and input as under operational conditions, that is *identification should be conducted in closed-loop.* Furthermore, the closed-loop data should be filtered properly in order to obtain the optimal model (in the sense of minimizing the criterion $J_1(\mathcal{D})$).

Hakvoort *et al.* also suggest to combine this identification strategy for a fixed controller with a control design method in an iterative manner. This is discussed in more detail in Section 5.7.1.

5.6.4 Control-relevant identification for GPC

A control-relevant prefilter for the GPC controller is developed in (Shook *et al.* 1991) and (Shook *et al.* 1992). They arrive at a FIR-filter of low-pass character obtained from a spectral factorization problem. Further comments and an application to a nonlinear multivariable anaesthetic model are given in (Mahfouf and Linkens 1994). In view of the current interest in predictive control algorithms we review the results below.

Consider, the simplified GPC-criterion

$$J_{GPC} = \sum_{k=N_1}^{N_2} [\hat{y}(t+kT) - w(t+kT)]^2 \tag{5.111}$$

where control weighting has been omitted. At each sample, the GPC controller calculates a control signal sequence that minimizes this sum of squared predicted control errors, i.e. errors between the predicted output and the reference. Hence, the the model used in the GPC algorithm should supply the best possible output predictions in the range N_1 to N_2 samples ahead. A standard least squares algorithm minimizes the variance of the one-step-ahead prediction. If the model set includes the true system then a good one-step prediction implies good long range predictions. However, in the presence of unmodelled dynamics a standard prediction error method can lead to models, which give very poor predictions further ahead. This will later be illustrated by simulations. Using an estimation algorithm that produces models which gives better predictions in the range N_1 to N_2 samples ahead may therefore be expected to significantly reduce the effects of unmodelled dynamics in the overall control design.

In pursuit of a more suitable estimation algorithm we make the following considerations, which were presented in (Shook *et al.* 1991, Shook *et al.* 1992). Consider the criterion

$$J_{des} = \sum_{k=N_1}^{N_2} [y(t+kT) - w(t+kT)]^2 \tag{5.112}$$

which can be perceived as the desired GPC control objective. The future output $y(t+kT)$ is of course unknown. It can be expressed as

$$y(t+kT) = \hat{y}(t+kT|t,\theta) + \varepsilon_k(t+kT,\theta) \tag{5.113}$$

where $\varepsilon_k(t+kT,\theta)$ is the k-step prediction of $y(t+kT)$ given a parameterized model with parameter vector θ. Insertion gives

$$\begin{aligned} J_{des} &= \sum_{k=N_1}^{N_2} [\hat{y}(t+kT|t,\theta) + \varepsilon_k(t+kT,\theta) - w(t+kT)]^2 \\ &= \sum_{k=N_1}^{N_2} [\hat{y}(t+kT|t,\theta) - w(t+kT)]^2 + \sum_{k=N_1}^{N_2} [\varepsilon_k(t+kT,\theta)]^2 \\ &\qquad + 2\sum_{k=N_1}^{N_2} [\hat{y}(t+kT|t,\theta) - w(t+kT)]\, \varepsilon_k(t+kT,\theta) \\ &= J_{GPC} + J_{ID} + J_{cross} \end{aligned} \tag{5.114}$$

where

$$J_{ID} = \sum_{k=N_1}^{N_2} [\varepsilon_k(t+kT,\theta)]^2 \tag{5.115}$$

and

$$J_{cross} = 2\sum_{k=N_1}^{N_2} [\hat{y}(t+kT|t,\theta) - w(t+kT)]\, \varepsilon_k(t+kT,\theta) \tag{5.116}$$

We see, that the desired criterion can be rewritten to a sum of three terms:

- J_{GPC} which is the usual GPC criterion.
- J_{ID} which is a sum of squared k-step prediction errors. This term should be minimized by the estimation algorithm.
- A cross term J_{cross} between the predicted control errors and the prediction errors. As argued in (Shook *et al.* 1992) this term can under mild conditions be considered to be zero on average. Especially, if a regression estimation method is used since then the prediction errors will be close to uncorrelated with the predictions.

In the following we will neglect the effects of the cross term. An estimation algorithm, which minimizes J_{ID} may be called a multi-step prediction error method, (Söderström and Stoica 1989).

Multi-step prediction estimation

To find a suitable multi-step prediction estimation algorithm we first set up a criterion function as in (Shook *et al.* 1992). In the multi-step prediction criterion we need to minimize a sum of squared k-step prediction errors rather than just the one-step prediction error. We extend the usual identification criterion (5.19) as follows.

$$\begin{aligned} J_{LRPI}(\theta) &= \frac{1}{N-N_2}\sum_{t=1}^{N-N_2}\sum_{k=N_1}^{N_2}\left[\varepsilon_k(t+kT,\theta)\right]^2 \qquad (5.117)\\ &= \frac{1}{N-N_2}\sum_{t=1}^{N-N_2}\sum_{k=N_1}^{N_2}\left[y(t+kT)-\hat{y}(t+kT|t,\theta)\right]^2 \end{aligned}$$

where N is the number of collected data. One way to proceed in developing a recursive algorithm to minimize such a criterion is to seek a direct minimization by solving numerically

$$\frac{\partial J_{LRPI}(\theta)}{\partial\hat{\theta}} = 0 \qquad (5.118)$$

or

$$\frac{1}{N-N_2}\sum_{t=1}^{N-N_2}\sum_{k=N_1}^{N_2}\left[y(t+kT)-\hat{y}(t+kT|t,\theta)\right]\frac{\partial\hat{y}(t+kT|t,\theta)}{\partial\hat{\theta}}=0$$

Such a problem can be attacked by a Gauss-Newton minimization procedure. This has been examined in (Lauritsen 1994) where a recursive procedure was developed. A problem is the highly nonlinear dependence between the parameter estimate and the gradient of the k-step predictors. This leads to poor and unknown convergence properties of the recursive algorithm. Another approach is to use a one-step prediction error method with a prefilter that asymptotically makes the algorithms equivalent.

Multi-step prediction estimation as equivalent prefilter

In Section 5.3 we saw, that using k-step prediction error methods in estimation changes the bias-distribution in a way similar to prefiltering, at least with a fixed noise model. This indicates, that the multi-step estimation algorithm can be replaced by an equivalent prefilter, that asymptotically gives the same parameter estimates. Such a prefilter was found in (Shook *et al.* 1991, Shook *et al.* 1992) as shown below.

Consider a true system on ARMAX form

$$A(q^{-1})y(t)=q^{-d}B(q^{-1})u(t)+C(q^{-1})e(t) \tag{5.119}$$

and an estimated system model

$$\hat{A}(q^{-1})y(t)=q^{-d}\hat{B}(q^{-1})u(t)+C(q^{-1})\xi(t) \tag{5.120}$$

Note, that we assume knowledge of the time-delay d of the system and of the noise polynomial $C(q^{-1})$ which is often considered a design parameter. The deterministic and the noise transfer functions of the true system are given by

$$G_0(q)=q^{-d}\frac{B(q^{-1})}{A(q^{-1})} \qquad H_0(q)=\frac{C(q^{-1})}{A(q^{-1})} \tag{5.121}$$

Using similar methods to those presented in Chapter 3 we can express the k-step prediction as

$$\hat{y}(t+kT|t,\theta) = \frac{F_k(q^{-1})}{C(q^{-1})}y(t) + \frac{E_k(q^{-1})\hat{B}(q^{-1})}{C(q^{-1})}u(t+(k-d)T) \quad (5.122)$$

where $E_k(q^{-1})$ and $F_k(q^{-1})$ are solutions to the Diophantine-equation

$$C(q^{-1}) = E_k(q^{-1})\hat{A}(q^{-1}) + q^{-k}F_k(q^{-1}) \quad (5.123)$$

where $\deg(E_k) = k-1$ and $\deg(F_k) = n-1$. This can be rearranged to

$$1 - q^{-k}\frac{F_k(q^{-1})}{C(q^{-1})} = \frac{E_k(q^{-1})\hat{A}(q^{-1})}{C(q^{-1})} \quad (5.124)$$

Insertion yields the k-step prediction error

$$\begin{aligned}
\varepsilon_k(t+kT) &= y(t+kT) - \frac{F_k}{C}y(t) + \frac{E_k\hat{B}}{C}u(t+(k-d)T) \\
&= \left(1 - q^{-k}\frac{F_k}{C}\right)y(t+kT) + \frac{E_k\hat{B}}{C}u(t+(k-d)T) \\
&= \frac{E_k\hat{A}}{C}y(t+kT) + \frac{E_k\hat{B}}{C}u(t+(k-d)T) \\
&= \frac{E_k\hat{A}}{C}\left(y(t+kT) - \frac{\hat{B}}{\hat{A}}u(t+(k-d)T)\right) \quad (5.125)
\end{aligned}$$

where dependence on θ and q^{-1} has been omitted for convenience. Inserting the true system equation in the above yields

$$\varepsilon_k(t+kT) = \frac{E_k\hat{A}}{C}\left[\left(\frac{B}{A} - \frac{\hat{B}}{\hat{A}}\right)u(t+(k-d)T) + \frac{C}{A}e(t+kT)\right] \quad (5.126)$$

Insertion into the multi-step prediction cost function (5.117) now gives

$$\begin{aligned}
J_{LRPI} = & \frac{1}{N-N_2}\sum_{t=0}^{N-N_2}\sum_{k=N_1}^{N_2}\left\{\left[\left(\frac{B}{A} - \frac{\hat{B}}{\hat{A}}\right)u(t+(k-d)T)\right.\right. \\
& \left.\left. + \frac{C}{A}e(t+kT)\right]\frac{E_k\hat{A}}{C}\right\}^2 \quad (5.127)
\end{aligned}$$

Using a technique similar to that in Section 5.3 it can be shown, that in open-loop the minimization of this criterion is asymptotically ($N \to \infty$) equivalent to minimizing

$$\bar{J} = \int_{-\pi}^{\pi} \sum_{k=N_1}^{N_2} \left[\left| \frac{B}{A} - \frac{\hat{B}}{\hat{A}} \right|^2 \Phi_u + \Phi_v \right] \left| \frac{E_k \hat{A}}{C} \right|^2 d\omega \tag{5.128}$$

with

$$\Phi_v = \left| \frac{C}{A} \right|^2 \Phi_e \tag{5.129}$$

The only quantity in (5.128) that depends on k is E_k. By comparing these expression with the results given in Section 5.3, in particular (5.40), we see that the multi-step prediction method asymptotically corresponds to a one-step-ahead prediction error method with a prefilter determined by

$$|L(e^{i\omega})|^2 = \sum_{k=N_1}^{N_2} |E_k(e^{i\omega})|^2 \tag{5.130}$$

or equivalently

$$\boxed{L(q)L(q^{-1}) = \sum_{k=N_1}^{N_2} E_k(q)E_k(q^{-1}) \qquad (5.131)}$$

This is a spectral factorization problem for which a number of possible solution methods exist. Note, that $L(q^{-1})$ is a finite impulse response filter of order $N_2 - 1$. This equivalent prefilter approach emulates a control-relevant identification problem. The bandwidth of the prefilter is automatically selected to fit the GPC criterion.

A simple factorization method, which essentially consists of repeated deconvolution is the Peterka's algorithm, (Mahfouf and Linkens 1994). Let

$$\Sigma_E = \sum_{k=N_1}^{N_2} E_k(q)E_k(q^{-1}) \tag{5.132}$$

We wish to find an L such that $L(q)L(q^{-1}) = \Sigma_E$. Suppose a recursive formula at step i has given the filter $L_i(q^{-1})$. Then the filter at step $i+1$ can be determined from

$$L_{i+1}(q)L_i(q^{-1}) \approx \Sigma_E \tag{5.133}$$

which gives the recursion

$$L_{i+1}(q) = \frac{\Sigma_E}{L_i(q^{-1})} + R_{i+1} \tag{5.134}$$

where R_i is the remainder at step i. Testing the amplitude of the numerically largest coefficient in R_i (the max-norm of R_i) can be used as a possible stop criterion.

The convergence of $L(q^{-1})$ is usually quite fast. To save computation time in recursive estimation a single iteration of (5.134) can be performed at each sample, of course starting from the result of the previous sample. This gives a very simple algorithm with only minimal extra computational burden as compared to standard RLS with prefiltering.

Example 9 *Open-loop simulation example*

Consider the true system

$$G(s) = \frac{2 \cdot 229}{(s+1)(s^2+30s+229)} \tag{5.135}$$

This is one of the so-called Rohrs examples, see (Rohrs et al. *1984). The system is sampled with a sampling period* $T = 0.05$*. We estimate a first order ARX model*

$$(1 + \hat{a}_1 q^{-1})y(t) = \hat{b}_0 u(t-T) + \epsilon(t) \tag{5.136}$$

using standard least-squares identification with no prefiltering. The input $u(t)$ *is a zero-mean Gaussian white noise sequence with variance* 1*. The experiment length is 2000 samples. Using this model we design with* $N_1 = 1$

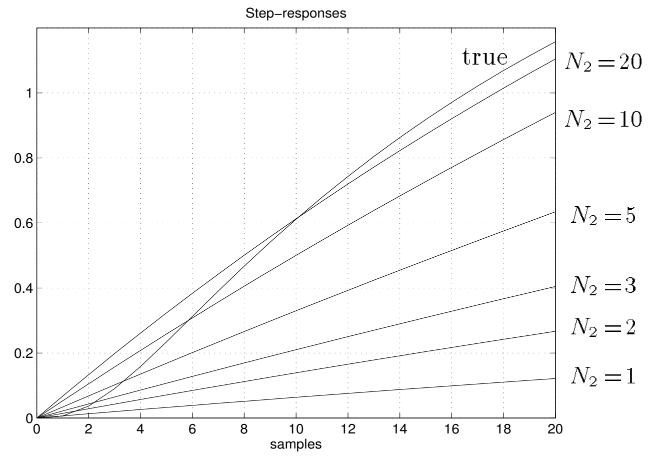

Figure 5.5: Step responses of estimated models with varying N_2.

and varying N_2 *filters using the equivalent prefilter approach described above. For each of these filters we estimate a new model of the system. The step-responses of the true system and estimated models are shown in Figure 5.5. Note, that for* $N_2 = 1$ *corresponding to no prefiltering the response of the model is very different from the true system and gives very misleading long range predictions. For all models the overall predictions are good in the range* N_1 *to* N_2 *samples ahead. The amplitudes of the equivalent prefilters are shown in Figure 5.6. The gains have been normalized to a DC-gain of 0 dB. The filters have low-pass characteristics with normalized break frequencies ranging from* $0.2 - 2$*, the higher* N_2 *the lower the break frequency.* □

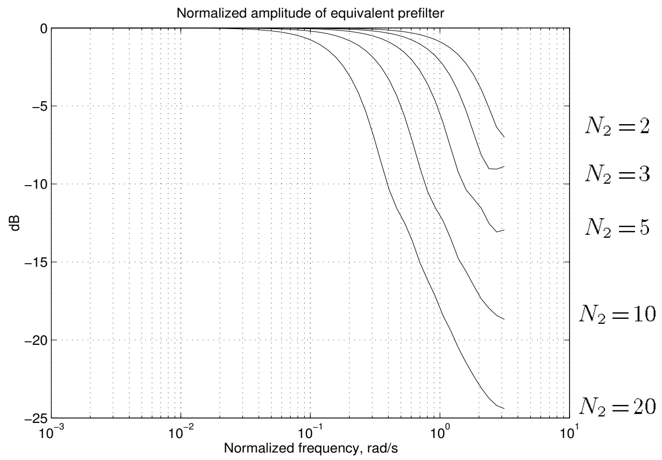

Figure 5.6: Amplitude of equivalent prefilters with varying N_2.

5.7 Iterative control design and system identification

In this section we consider iterative approaches to control design and system identification as motivated in Section 5.1. Extensive introductions to iterative procedures of combining system identification and control design are given in (Gevers 1993, Van den Hof and Schrama 1995).

Suppose, that we want to minimize some norm of a control performance cost $J(G_0, C)$ where G_0 is the true plant and C is the controller with a given structure. Let $\hat{G}(\theta)$ denote a parameterized model of G_0, where θ is the parameter vector. In model-based control design the controller is a function of this model, i.e. $C = C(\hat{G}(\theta))$. The ideal approach to solving this problem can then be stated as the minimization problem

$$\hat{\theta}_{opt} = \arg\min_{\theta} \left\| J\left(G_0, C(\hat{G}(\theta))\right) \right\| \qquad (5.137)$$

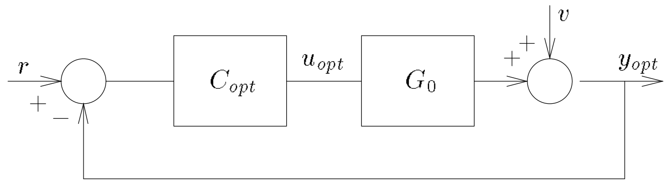

Figure 5.7: Optimal closed-loop.

$$C_{opt} = C(\hat{G}(\hat{\theta}_{opt})) \tag{5.138}$$

where C_{opt} denotes the optimal controller within the given controller structure. The above problem is a joint problem of system identification and control design. The resulting optimal closed-loop is depicted in Figure 5.7. Typically this problem is intractable because it depends on the true plant. Using the certainty equivalence principle we would replace the above minimization problem by

$$\hat{\theta} = \arg\min_{\theta} \|J(\hat{G}(\theta), C(\hat{G}(\theta)))\| \tag{5.139}$$

$$C = C(\hat{G}(\hat{\theta})) \tag{5.140}$$

Again we have a joint problem of identification and control. Such problems are usually not feasible in practice.

One way to link identification and control is the following. With a given controller $C = C(\hat{G})$ designed for the model $\hat{G}$ consider the triangle inequality (Schrama 1992)

$$\begin{aligned} \Big| \|J(\hat{G}, C)\| - \|J(G_0, C) - J(\hat{G}, C)\| \Big| &\leq \|J(G_0, C)\| \\ &\leq \|J(\hat{G}, C)\| + \|J(G_0, C) - J(\hat{G}, C)\| \end{aligned} \tag{5.141}$$

The term $\|J(G_0, C)\|$ is the *achieved performance* of the real plant controlled by C, i.e. the performance in the achieved or obtained closed-loop, see Figure 5.8. The term $\|J(\hat{G}, C)\|$ is the *designed performance*, which is evaluated in the designed or nominal closed-loop, see Figure 5.9. Finally

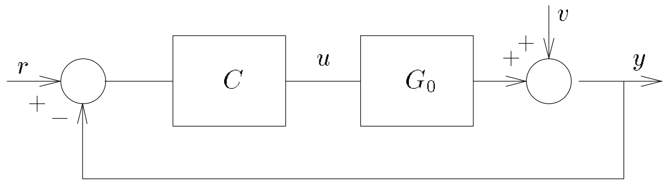

Figure 5.8: Achieved closed-loop.

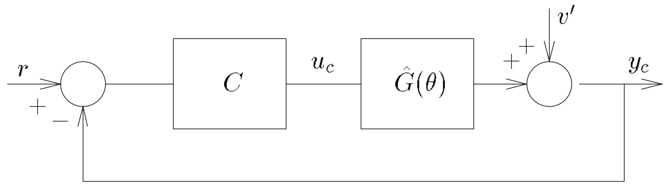

Figure 5.9: Designed closed-loop.

$\|J(G_0, C) - J(\hat{G}, C)\|$ is the *performance degradation* due to the fact, that the controller C has been designed for the estimated plant $\hat{G}$ and not the true plant G_0.

The objective of the control design is ideally to make the designed closed-loop behave as closely as possible to the optimal closed-loop. However, as C_{opt} is unknown, there is no way to evaluate the closeness of these two loops. The best alternative is to compare the designed loop with the achieved loop. The triangle equality (5.141) shows, that the designed loop can be made close to the achieved loop by minimizing the performance degradation $\|J(G_0, C) - J(\hat{G}, C)\|$. One may argue, that this is exactly the goal of a separate identification step. In this way we have found a truly control-relevant identification problem directly induced by the control criterion, (Van den Hof and Schrama 1995).

Instead of considering the intractable joint minimization of the control criterion one may attempt to attack the problem iteratively, i.e. with a fixed model $\hat{G}(\hat{\theta}_i)$ find a controller by

$$C_i = \arg\min_C \|J(\hat{G}(\hat{\theta}_i), C)\| \tag{5.142}$$

This is simply a model-based control design. Apply this controller and obtain new input-output data. At the next step find the new model by the identification step

$$\hat{\theta}_{i+1} = \arg\min_\theta \|J(G_0, C_i) - J(\hat{G}(\theta), C_i)\| \tag{5.143}$$

and the next controller by the control design step

$$C_{i+1} = \arg\min_C \|J(\hat{G}(\hat{\theta}_{i+1}), C)\| \tag{5.144}$$

By continuing this iterative procedure one may hope to asymptotically get close to a solution of the ideal joint problem (5.137). We will call the overall control objective $J(G_0, C)$ the *global problem*. The successive steps of the iterative approach are *local* problems in the sense, that either the model or the controller is fixed in the minimization problem.

One difficulty of the iterative procedure is the identification step (5.143). This is very likely to be a nonstandard identification problem not easily solved. Some different procedures have been considered to overcome this difficulty for various control criteria.

Rather than finding alternative nonstandard control-relevant identification strategies one may find it useful to take starting point in a known feasible identification method, for instance a prediction error method. By designing the datafilter properly it may be possible to asymptotically approximate other control-relevant identification strategies. This is done in the Zang-scheme described in Section 5.7.3.

General iterative scheme

In view of the above discussion a general iterative scheme can be proposed, see (Gevers 1993).

1. Find, for instance by open-loop identification, an initial model $\hat{G}_0$ from input-output data.

2. Design a controller C_i. Apply C_i to the plant and collect closed-loop data.

3. Identify a new model $\hat{G}_{i+1}$ from a local identification problem.

4. Iterate to 2 with $i := i + 1$ as long as necessary.

The number of iterations is not supposed to be infinite for two reasons (Gevers 1993): First, for most iterative schemes no convergence proofs exist and secondly, for many practical problems there is no need to update the controller frequently. Hence, iterate until satisfactory performance is obtained. Time-varying systems is a problem, see Section 5.8.

In the following we will present several approaches to iterative identification and control, which are variants of the general scheme. They consider both nonstandard identification problems and approximation via control-relevant prefiltering.

5.7.1 Iterative approaches with control-relevant prefiltering

An obvious approach to an iterative identification and control scheme with control-relevant prefilters is depicted in Figure 5.10. Such a scheme was suggested in (Hakvoort *et al.* 1994) in connection with the iterative prefiltering design discussed in Section 5.6.3.

A specialty of this iterative approach is the inner loop for iterative prefilter design. The idea is to refine the prefilter based on the same set of closed-loop data by repeated prefiltering and estimation. The prefilter is updated

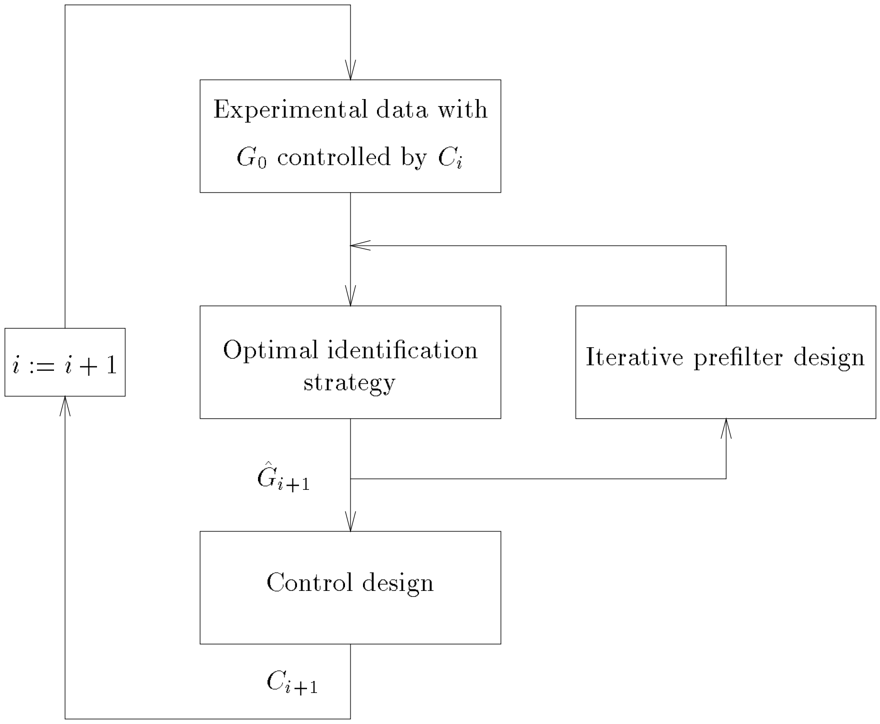

Figure 5.10: Iterative identification and control scheme with control-relevant prefiltering.

at each iteration of the inner loop based on the estimated model of the previous iteration.

The combination of control-relevant prefiltering and control design is considered in many papers, see for instance (Åström 1993, Åström and Nilsson 1994, Vodă and Landau 1994, Vodă and Landau 1995, Hakvoort *et al.* 1994, Zang *et al.* 1995).

We present two examples on this approach using the GPC equivalent prefilter approach described in Section 5.6.4 and the optimal choice of prefilter described in Section 5.6.1.

Example 10 *Consider as in Example 9 the true system*

$$G(s) = \frac{2 \cdot 229}{(s+1)(s^2+30s+229)} \tag{5.145}$$

sampled with $T = 0.05$ *to give a true system of the form*

$$A(q^{-1})y(t) = q^{-1}B(q^{-1})u(t) + e(t) \tag{5.146}$$

The noise sequence is zero-mean Gaussian white noise with variance 0.005. *The control design model is a first order ARIMAX model*

$$(1 + \hat{a}_1 q^{-1})y(t) = \hat{b}_0 u(t-T) + \frac{T(q^{-1})}{1-q^{-1}}\xi(t) \tag{5.147}$$

Consider the basic GPC controller as described in Section 4.2.1 with design parameters

$$N_1 = 1, \quad N_2 = 20, \quad N_u = 0, \quad \rho = 0 \tag{5.148}$$

and an observer polynomial

$$T(q^{-1}) = 1 - 0.85q^{-1} \tag{5.149}$$

Using an open-loop estimation of the initial model with no prefiltering corresponding to the case $N_1 = N_2 = 1$ *in Example 9 results in a destabilizing controller. Instead we arbitrarily choose the initial model* $\hat{a}_1 = -0.7$,

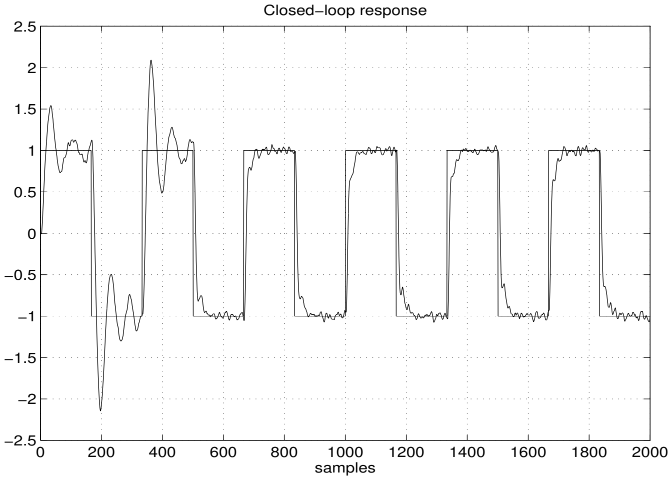

Figure 5.11: Closed-loop response obtained in Example 10, iterative update of GPC and equivalent prefilter.

$\hat{b}_0 = 0.5$. *We now run the iterative procedure as described above with a total experiment length of 2000 samples. The reference signal is a square-wave. The model estimate, prefilter and controller are updated in that order every 500 samples. Estimation is based on data from the previous 500 samples. We get the closed-loop responses shown in Figure 5.11. We observe a very fast convergence of the iterative algorithm. Actually, no improvement can be seen already after the first iteration. The amplitude characteristic of the obtained prefilter after 2000 samples is shown in Figure 5.12. The obtained prefilter is almost identical to the one found in Example 9 in the case* $N_2 = 20$, *see Figure 5.6.* □

Example 11 *Consider the same experimental conditions as in Example 10. This time we use the prefilter (5.87) described in Section 5.6.1. The obtained closed-loop response is shown in Figure 5.13. The responses are very similar to those obtained with the equivalent GPC prefilter. The amplitude characteristic of the obtained prefilter after 2000 samples is shown in Figure 5.14.*

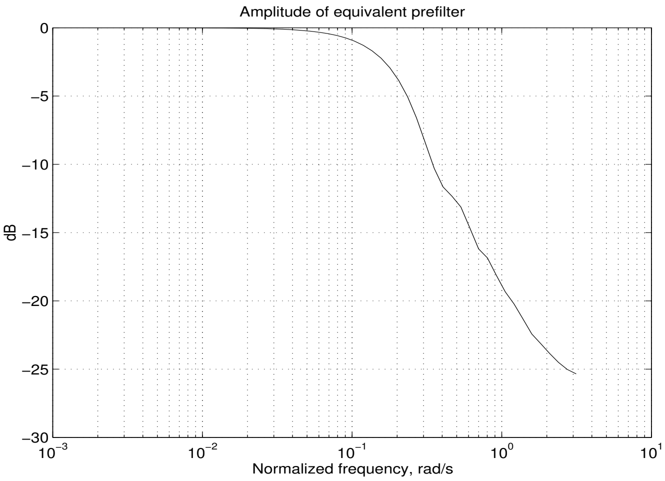

Figure 5.12: Amplitude of equivalent prefilter obtained in Example 10.

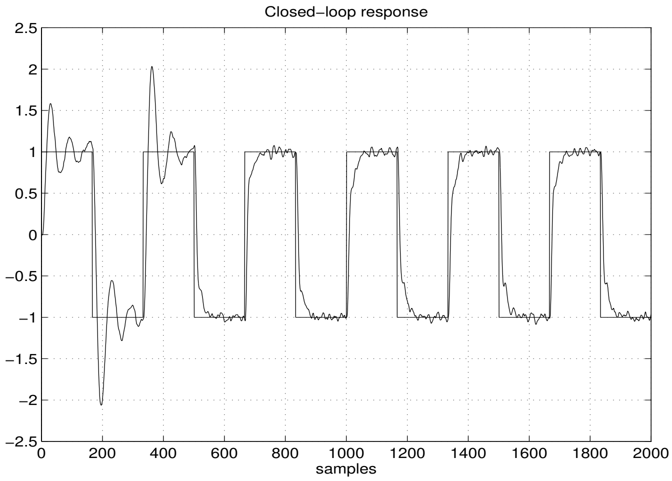

Figure 5.13: Closed-loop response obtained in Example 11, iterative update of GPC and prefilter (5.87).

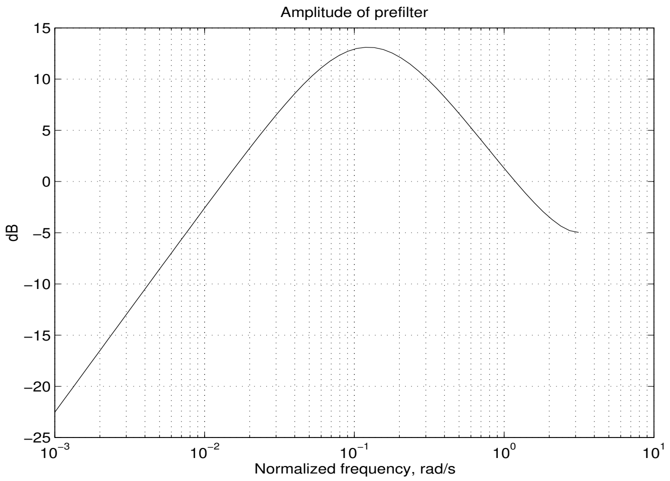

Figure 5.14: Amplitude of obtained prefilter in Example 11.

This prefilter has approximately the same bandwidth as the filter shown in Figure 5.12 but it has band-pass characteristics. It is a common observation in ad hoc prefilter design, that frequencies around the closed-loop bandwidth should be emphasized, see (Mohtadi 1988). □

5.7.2 The Schrama scheme

One of the first iterative model-based control design schemes where the identification criterion is derived directly from the control criterion is (Schrama 1992, Schrama and Van den Hof 1992). As in (Gevers 1993) we name it the Schrama scheme. A feedback configuration as shown in Figure 5.15 is considered. The signals r_1 and r_2 are reference or probing signals and v is a disturbance signal.

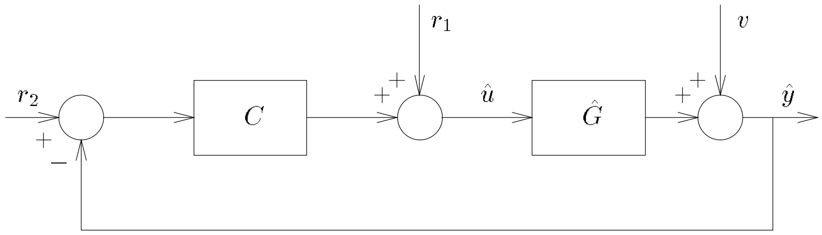

Figure 5.15: Feedback configuration in the Schrama scheme.

Control design

The model-based controller is found as an H_∞ design based on robustness optimization via the minimization

$$C_{\hat{G}} = \arg\min_{C} \|T(\hat{G}, C)\|_\infty \tag{5.150}$$

where $T(\hat{G}, C)$ is the transfer matrix

$$T(\hat{G}, C) = \begin{bmatrix} \hat{G}(I + \hat{G}C)^{-1}C & \hat{G}(I + \hat{G}C)^{-1} \\ (I + \hat{G}C)^{-1}C & (I + C\hat{G})^{-1} \end{bmatrix} \tag{5.151}$$

so that

$$\begin{bmatrix} \hat{y} \\ \hat{u} \end{bmatrix} = T(\hat{G}, C) \begin{bmatrix} r_2 \\ r_1 \end{bmatrix} \tag{5.152}$$

The performance of the actual plant G_0 can be investigated by means of the triangle equality

$$\|T(G_0, C_{\hat{G}})\|_\infty \leq \|T(\hat{G}, C_{\hat{G}})\|_\infty + \|T(G_0, C_{\hat{G}}) - T(\hat{G}, C_{\hat{G}})\|_\infty \tag{5.153}$$

where we identify

1. $\|T(G_0, C_{\hat{G}})\|_\infty$ the *achieved performance.*
2. $\|T(\hat{G}, C_{\hat{G}})\|_\infty$ the *designed performance.*

3. $\|T(G_0, C_{\hat{G}}) - T(\hat{G}, C_{\hat{G}})\|_\infty$ the *performance degradation*.

The designed performance is minimized in the control design minimization problem given a plant estimate $\hat{G}$. To keep the designed performance close to the achieved performance it is desirable to minimize the performance degradation in the approximation/estimation phase. It appears, that the performance degradation depends on both the plant estimate and the controller $C_{\hat{G}}$ found from this estimate. Hence, the minimization of the performance degradation and the designed performance needs to be treated as a joint problem. Schrama suggests the following iterative procedure:

1. In the ith step obtain data from the plant operating under feedback with controller C_{i-1}.
2. Find the nominal model $\hat{G}_i$ according to
$$\hat{G}_i = \arg\min_{\overline{G} \in \mathcal{G}} \|T(G_0, C_{i-1}) - T(\overline{G}, C_{i-1})\|_\infty \tag{5.154}$$
where $\mathcal{G}$ is a set of candidate models.
3. Find C_i according to
$$C_i = \arg\min_{C} \|T(\hat{G}_i, C)\|_\infty \tag{5.155}$$
4. Apply C_i to the plant and set $i := i + 1$.

The flow of this iterative scheme is illustrated in Figure 5.16.

Identification scheme

As there exists no identification technique to solve the problem (5.154) the L_∞ approximation is replaced by an L_2 approximation so that based on

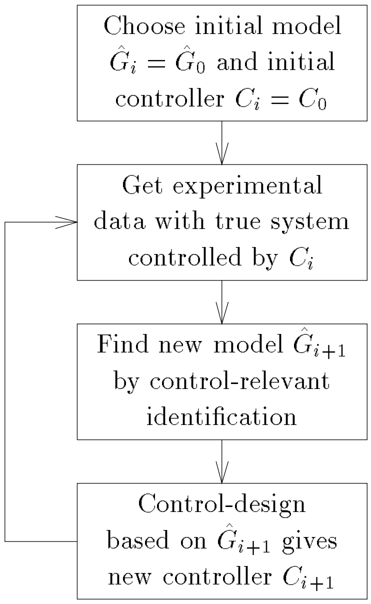

Figure 5.16: Iterative identification and control according to (Schrama 1992).

closed-loop data of the plant input u and output y the model is determined as

$$\hat{G}_i = \arg\min_{\overline{G}\in\mathcal{G}} \|T(G_0, C_{i-1}) - T(\overline{G}, C_{i-1})\|_2 \tag{5.156}$$

This replacement is backed up by the results presented in (Caines and Baykal-Gürsoy 1989) on L_∞ consistency of L_2 estimators. In other words, the ∞-norm of the performance degradation is reduced by minimizing its 2-norm. The problem (5.156) is still a non-standard identification problem.

In (Schrama 1992) the identification problem is solved by means of a coprime factorization technique. Assume, that the plant G_0 is stabilized by the controller C_{i-1}. Since C_{i-1} is known from the previous iteration it can be used to parameterize the set of all stabilized systems by their right coprime factorizations—one of these is the true plant G_0

$$G_0 = (N_a + D_c R)(D_a - N_c R)^{-1} \tag{5.157}$$

where R is a stable transfer function, $C_{i-1} = N_c D_c^{-1}$ and we choose $\hat{G}_a = N_a D_a^{-1}$ as some auxiliary model stabilized by C_{i-1}. The pairs (N_c, D_c) and (N_a, D_a) are coprime factorizations, where N_c, D_c, N_a and D_a are all stable transfer functions. Define

$$N^a = N_a + D_c R \tag{5.158}$$

$$D^a = D_a - N_c R \tag{5.159}$$

so that

$$G_0 = N^a (D^a)^{-1} \tag{5.160}$$

Note, that N^a and D^a are unknown, because the particular R for which the above holds is unknown. Furthermore define the signal x where $y = N^a x$ and $u = D^a x$. (Schrama 1992) shows, that x can be reconstructed by

$$x = (D_a + C_{i-1} N_a)^{-1}(u + C_{i-1} y) \tag{5.161}$$

From Figure 5.15 it appears, that $u + C_{i-1} y = C r_2 + r_1$. Hence x is uncorrelated with the noise signal v, provided that v is uncorrelated with r_1 and

r_2. The closed-loop identification problem has now been transformed to an open-loop one.

As both D_a, N_a and C_{i-1} are known, the frequency response of N^a and D^a can be estimated from $y = N^a x$ and $u = D^a x$. With these frequency response estimates an estimate of the frequency response of $T(G_0, C_{i-1})$ can be constructed. It now remains to estimate $\hat{G}_i$ from (5.156). As noted in (Schrama 1992) this frequency domain identification problem is not trivial as the criterion function minimized in (5.156) is nonlinear and with a complicated dependence on $\hat{G}_i$.

More recent work on the solution of the identification problem for the Schrama scheme can be found in (Van den Hof *et al.* 1993, de Callafon and Van den Hof 1995b, Van den Hof *et al.* 1995). Some work on solving the H_∞ identification problem (5.154) has been presented in (de Callafon and Van den Hof 1995a) where a stable factorization of the model is being fit by pointwise evaluation of the error at a prespecified set of frequencies. They arrive at a non-linear constrained optimization problem with a convex parameter space. Such a problem can be solved by means of standard minimization routines.

Cautious controller update

As a refinement to the basic iterative scheme (Schrama and Van den Hof 1992, Schrama and Bosgra 1993) suggests a cautious controller update. Consider the control optimization problem

$$C_i = \arg\min_C \|T(\alpha_i \hat{G}_i, C/\alpha_i)\|_\infty \tag{5.162}$$

where α_i is an adjustable scalar weight. As noted in (Schrama and Van den Hof 1992) the closed-loop bandwidth of the resulting closed-loop system will be close to the cross-over frequency of $\alpha_i \hat{G}_i$. Hence, a large α_i corresponds to a high nominal performance. The idea is then to gradually increase α_i

during the iterations. Guidelines for the choice of α_i are given in (Schrama and Van den Hof 1993). As the identification criterion (5.154) is derived from the control criterion it should be modified accordingly to

$$\hat{G}_i = \arg\min_{\overline{G}\in\mathcal{G}} \|T(\alpha_i G_0, C_{i-1}/\alpha_i) - T(\alpha_i \overline{G}, C_{i-1}/\alpha_i)\|_\infty \tag{5.163}$$

5.7.3 The Zang-scheme

One of the most interesting algorithms, which iteratively combine system identification and control design in a sensible way is the so-called Zang-scheme (Zang *et al.* 1995). The Zang-scheme provides two important extensions to model-based control design.

- Control-relevant prefiltering with the prefilter determined in accordance with the control criterion.
- Frequency-weighted control design where the control weights are chosen in accordance with closed-loop data. This can be seen as a way of bypassing the certainty equivalence principle.

The introduction of a frequency weighted control design as an integral part of the iterative identification and control scheme is a novelty. The algorithm was initially proposed in (Zang *et al.* 1991) for the tracking problem and (Zang *et al.* 1992) for the disturbance rejection problem. Here we consider the disturbance rejection problem as described in (Zang *et al.* 1995).

Model description

Consider a true plant with input-output description

$$y(t) = G_0(q)u(t) + v(t) \tag{5.164}$$

$$v(t) = H_0(q)e(t) \tag{5.165}$$

where $G_0(q)$ is a strictly proper transfer function, $u(t)$ is the system input, $e(t)$ is a white noise signal and $v(t)$ is an unmeasurable disturbance acting on the input modelled as filtered white noise. The true plant is modelled by a parameterized model

$$y_c(t,\theta) = \hat{G}(q,\theta)u_c(t,\theta) + v'(t) \tag{5.166}$$

$$v'(t) = \hat{H}(q)e'(t) \tag{5.167}$$

where θ is the parameter vector. The noise model $\hat{H}(q)$ is fixed by the designer.

Control design

Consider the global LQG control criterion for disturbance rejection

$$J_{\text{global}} = \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N}(y(t)^2 + \lambda^2 u(t)^2) \tag{5.168}$$

Using the usual certainty equivalence approach one would replace the global criterion with

$$\overline{J} = \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{N}\left[(y_c(t))^2 + \lambda^2(u_c(t))^2\right] \tag{5.169}$$

where $y_c(t)$ is the output of the identified model $\hat{G}(q,\theta)$ driven by the input $u_c(t)$. Recognizing that the control design is based on an identified model of the true plant while being applied to the true model Zang *et al.* choose to consider the following frequency weighted local criterion

$$J_C = \lim_{N\to\infty} \frac{1}{N}\sum_{t=1}^{\infty}(F_1(q)y_c(t))^2 + \lambda^2(F_2(q)u_c(t))^2 \tag{5.170}$$

where F_1 and F_2 are linear filters to be chosen. By comparing the global criterion with the above one realizes, that the data filters should be chosen according to

$$F_1 = \left(\frac{\Phi_y}{\Phi_{y_c}}\right)^{\frac{1}{2}} \tag{5.171}$$

$$F_2 = \left(\frac{\Phi_u}{\Phi_{u_c}}\right)^{\frac{1}{2}} \tag{5.172}$$

where Φ_y, Φ_u, Φ_{y_c} and Φ_{u_c} are the spectra of the corresponding closed-signals with the previous controller C_i. These spectra are estimated by fitting low-order ARX-models to the data sequences. It is important to note, that data for these estimation problems is readily available from the actual closed-loop and the simulated closed-loop. Also note, that since $u(t) = -C_i(q)y(t)$ and $u_c(t) = -C_i(q)y_c(t)$ it holds, that

$$F_1 = F_2 = \left(\frac{\Phi_y}{\Phi_{y_c}}\right)^{\frac{1}{2}} = \left(\frac{\Phi_u}{\Phi_{u_c}}\right)^{\frac{1}{2}} \tag{5.173}$$

Choosing the data filters in this way makes the spectra of the filtered output and control signals in the designed closed-loop system equal to the spectra of the output and input of the actual closed-loop system. Hence, minimizing the local design criterion J_C effectively minimizes the global criterion J_{global}. Of course, the filters used in practice are approximations of these ideal filters.

Identification scheme

Introduce the vector 2-norm definition

$$\left\| \begin{array}{c} x_1(t) \\ x_2(t) \end{array} \right\|_2 \triangleq \sqrt{\sum_{t=1}^{N}\left[\left(\frac{1}{\sqrt{N}}x_1(t)\right)^2 + \left(\frac{1}{\sqrt{N}}x_2(t)\right)^2\right]} \tag{5.174}$$

With this definition the global control criterion for finite time can be expressed as

$$J^N_{\text{global}} = \left\| \begin{array}{c} y(t) \\ \lambda u(t) \end{array} \right\|_2^2 \tag{5.175}$$

The following triangle inequality can be set up

$$\begin{aligned} \left| \left\| \begin{array}{c} y_c(t) \\ \lambda u_c(t) \end{array} \right\|_2 - \left\| \begin{array}{c} y(t) - y_c(t) \\ \lambda(u(t) - u_c(t)) \end{array} \right\|_2 \right| &\leq \left\| \begin{array}{c} y(t) \\ \lambda u(t) \end{array} \right\|_2 \\ \leq \left\| \begin{array}{c} y_c(t) \\ \lambda u_c(t) \end{array} \right\|_2 &+ \left\| \begin{array}{c} y(t) - y_c(t) \\ \lambda(u(t) - u_c(t)) \end{array} \right\|_2 \end{aligned} \tag{5.176}$$

The first term of the upper bound is the performance achieved by the frequency unweighted design based on plant estimate $\hat{P}$, that is the controller coming from certainty equivalence considerations. The last term, which hence is a measure of the closed-loop model error is chosen as the local identification criterion

$$J^N_I = \frac{1}{N} \sum_{t=1}^{N} \left[(y(t) - y_c(t))^2 + \lambda^2 (u(t) - u_c(t))^2 \right] \tag{5.177}$$

Note, that this criterion measures the closed-loop model error in the same norm as used in the control criterion. This is different from the method proposed by Schrama, see Section 5.7.2. Asymptotically as $N \to \infty$, under weak conditions, the local identification criterion converges to

$$J_I = \frac{1}{2\pi} \int_{-\pi}^{\pi} (1 + \lambda^2 |C_i|^2) \left| \frac{H_0 - \hat{H} + (H_0 \hat{G} - \hat{H} G_0) C_i}{(1 + G_0 C_i)(1 + \hat{G} C_i)} \right|^2 d\omega \tag{5.178}$$

For $N \to \infty$ we now see, that the triangle equality can be expressed as

$$\left| (\overline{J})^{1/2} - (J_I)^{1/2} \right| \leq (J_{\text{global}})^{1/2} \leq (\overline{J})^{1/2} + (J_I)^{1/2} \tag{5.179}$$

Hence, by minimizing J_I in the identification step we force the certainty equivalence design criterion to get close to the global criterion.

Consider the usual filtered least squares identification criterion

$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}[\varepsilon_f(t,\theta)]^2 = \frac{1}{N}\sum_{t=1}^{N}\{L(q)\,[y(t) - \hat{y}(t|\theta)]\}^2 \tag{5.180}$$

where $\varepsilon_f(t,\theta)$ is the filtered one-step prediction error and $\hat{y}(t,\theta)$ is the one-step predictor associated with the model for a given parameter vector θ

$$\hat{y}(t|\theta) = \hat{H}^{-1}(q)\hat{G}(q,\theta)u(t) + \left[1 - \hat{H}^{-1}(q)\right]y(t) \tag{5.181}$$

so that

$$\varepsilon_f(t,\theta) = L(q)\,(y(t) - \hat{y}(t|\theta)) \tag{5.182}$$

Under reasonable conditions (Ljung 1987) $J_N(\theta)$ converges with closed-loop estimation in the limit $N \to \infty$ to (5.54)

$$J(\theta) = \int_{-\pi}^{\pi}\left|\frac{1}{1+G_0C_2}\right|^2\left[|G_0 - \hat{G}|^2|C_1|^2\Phi_r + \left|1+\hat{G}C_2\right|^2\Phi_v\right]\left|\frac{L}{\hat{H}}\right|^2 d\omega \tag{5.183}$$

as shown in Section 5.4.

Zang *et al.* prove the following result.

Theorem 17 *(Zang* et al. *1995) Identification for performance*

Let $e'(t) = e(t)$ and $\hat{H} = H_0$. Then with the reference signal spectrum

$$\Phi_r(\omega) = \gamma^2\Phi_v(\omega) = \gamma^2|H_0(e^{i\omega})|^2 \qquad \forall\omega \in [-\pi,\,\pi[\tag{5.184}$$

for some nonzero γ and with the data filter chosen as

$$L(q,\theta) = \hat{H}(q)D(q)\left[1 + \hat{G}(q,\theta)C_i(q)\right]^{-1} \tag{5.185}$$

where $D(q)$ is a stable spectral factor of $1 + \lambda^2 C_i(q)C_i^*(q)$ it holds that

$$\arg\min_\theta J_I = \arg\min_\theta J(\theta) \tag{5.186}$$

□

This theorem provides a control-relevant prefilter for the LQG regulator based on the control performance error criterion J_I. Note, that identification must take place in closed-loop. The choice of external excitation signal plays an important role in getting the desired result from the parameter estimation. It is also worth mentioning, that the filter $L(q,\theta)$ actually depends on the model $\hat{G}(q,\theta)$. In practice, $\hat{G}(q,\theta)$ in (5.185) is replaced by the model obtained at the previous iteration of the iterative scheme.

In cases where it is difficult to maintain closed-loop stability it is not desirable to identify models with respect to performance measures. (Zang *et al.* 1995) proposes another set of design variables where emphasis is put on robust stability rather than performance. The standard small-gain condition for robust stability with additive plant uncertainty is (Zhou *et al.* 1996)

$$\left\| \left(G_0 - \hat{G}\right) \frac{C}{1+G_0 C} \right\|_\infty < 1 \tag{5.187}$$

This is a sufficient condition. Zang *et al.* show the following result which provides an alternative to the strategy for identification for performance in cases where this fails to provide a robustly stable closed-loop.

Theorem 18 *(Zang* et al. *1995) Identification for robust stability*

With data filter

$$L(q) = \hat{H}(q) \tag{5.188}$$

and external excitation with spectrum

$$\Phi_r(\omega) \approx \gamma^2 \gg |\Phi_v(\omega)| \quad \forall\omega \in [-\pi,\, \pi\,[\tag{5.189}$$

it holds, that

$$\arg\min_\theta V(\theta) = \arg\min_\theta \left\| \frac{\left[G_0(e^{i\omega}) - \hat{G}(e^{i\omega},\theta)\right] C_i(e^{i\omega})}{1+G_0(e^{i\omega})C_i(e^{i\omega})} \right\|_2 \tag{5.190}$$

□

We see a clear resemblance between (5.190) and (5.187). One obvious difference is the fact, that the identification criterion is measured in a 2-norm while the robust stability condition requires an ∞-norm. This is of course not ideal but the use of ∞-norm identification on quite short data sequences is not useful in practice. Aside from this the development of ∞-norm identification algorithms is on a stage not ready for practical implementation.

Robust stability test

In order to detect when to identify for robust stability using the setup from Theorem 18 instead of performance using the setup in Theorem 17 Zang *et al.* describe a test for robust stability in (Zang *et al.* 1995). The idea is to directly identify accurate high-order ARX-models of the closed-loop sensitivity functions for achieved and designed loops using the procedure described in (Van den Hof and Schrama 1993). Using these estimates a stability test function

$$\phi_i(z) = \frac{(G_0 - \hat{G}_{i+1})C_{i+1}}{1 + \hat{G}_{i+1}C_{i+1}} \tag{5.191}$$

can be generated, compare with (5.187). Checking that the magnitude of this test function remain bounded by one for all frequencies is a sufficient condition for robust stability. However, it is well-known that such small-gain considerations may be quite conservative and Zang *et al.* also remarks, that in practice violation of this bound at a few points usually does not give instability of the achieved closed-loop.

Iterative procedure

The iterative procedure of the Zang-scheme proceeds as follows

1. *Compute data filter for identification and select excitation signal.* With $\hat{G}_i$, C_i, $\hat{H}$ and λ given select

   - Performance

   $$L_i = \frac{\hat{H}}{1 + \hat{G}_i C_i}(1 + \lambda^2 |C_i|^2)^{1/2}, \quad \Phi_r = \gamma^2 \Phi_v \tag{5.192}$$

   - Robust stability

   $$L_i = \hat{H}, \quad \Phi_r = \gamma^2 \tag{5.193}$$

   Using these experimental conditions generate closed-loop signals $\{y(t)\}$ and $\{u(t)\}$.

2. *Model identification.* Using least-squares prediction error identification compute $\hat{G}_{i+1}$ based on $\{y(t)\}$, $\{u(t)\}$, $\{r(t)\}$ and $L_i(q)$ from step 1.

3. *Compute controller data filters.* With controller C_i acting on G_0 and no external excitation signal ($r(t) = 0$) generate closed-loop signals and estimate a low-order AR-model $A_u^{-1}(q)$ for $\{u(t)\}$. With the controller acting in feedback with $\hat{G}_{i+1}$ simulate closed-loop data and estimate low-order AR-model $A_{u_c}^{-1}$ for $\{u_c(t)\}$. Compute the frequency weighting filter as

   $$F_i(q) = \frac{A_{u_c}(q)}{A_u(q)} \tag{5.194}$$

4. *Compute LQG controller.* With $\hat{G}_{i+1}$, $\hat{H}$ and F_i compute the new controller C_{i+1}.

5. *Test for robust stability.* Using high-order estimates of closed-loop sensitivity function for achieved loop with C_i, designed loop with $\hat{G}_{i+1}$ test for robust stability.

6. *Loop.* If robust stability test is passed then proceed to step 1 with performance objective; otherwise set $C_{i+1} := C_i$ and proceed to step 1 with robust stability objective.

Extensions

In the original version of the Zang-scheme presented in (Zang *et al.* 1991) the frequency weights used in the control design to obtain C_{i+1} were based on simulations with the old plant estimate $\hat{G}_i$ rather than $\hat{G}_{i+1}$. In (Partanen and Bitmead 1993) it is remarked, that this has the effect of getting the control design signals y_c and u_c out of synchronization with the frequency weight F_i. Partanen and Bitmead propose a so-called two-stage refinement where the newly identified model $\hat{G}_{i+1}$ is used to derive the frequency weight. More recent version of the Zang-scheme as in (Zang *et al.* 1995) utilizes this two-stage update, see step 3 of the above iterative procedure.

Another refinement proposed in (Partanen and Bitmead 1993) is to omit the plant estimate update in the so-called direct experimental controller refinement. The iterative update of the controller is maintained. Such an approach has the advantage, that the need of the external excitation signal $r(t)$ is removed. Of course, the performance may be expected to be worse than that of the two-stage refinement as no attempt is made to decrease the model discrepancy from the true plant. This is confirmed by simulation results in (Partanen and Bitmead 1993). However, there it is also shown by simulation, that a composite iterative design where the two-stage update of model and controller is used in the first few steps and then only the controller update using the direct controller refinement, actually achieves better performance at later iterations than the two-stage update. It is remarked, that this may be due to limitations imposed by undermodelling and/or proper selection of plant excitation signals.

An industrial application to a sugar cane crushing mill was presented in (Partanen and Bitmead 1995). A combination of the standard approach and the direct approach described above was used.

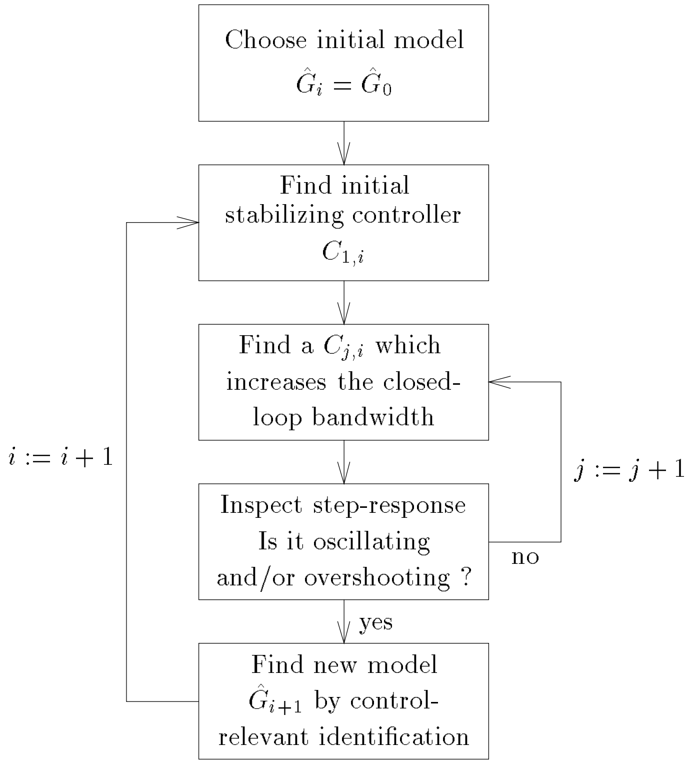

Figure 5.17: The windsurfer approach

5.7.4 The windsurfer approach

Another approach to iterative identification and control is taken in (Lee *et al.* 1992, Lee *et al.* 1993, Lee *et al.* 1995) where the so-called windsurfer approach is described. The general idea of this scheme is depicted in Figure 5.17.

It appears, that the windsurfer approach consists of an inner and an outer loop. The inner loop seeks to improve the performance by increasing the bandwidth of the closed-loop system through the design of more and more demanding controllers. However, at a certain point the discrepancy between the system model and the true system makes it impossible to improve the closed-loop performance. Hence, we escape to the outer loop where the model is improved by control-relevant system identification and then return to the inner loop.

Intuitively it seems reasonable, that the closed-loop bandwidth should be smoothly increased as more and more knowledge about the system is obtained. The name of the approach comes from an analogy with the learning of how to windsurf. Naturally, a newcomer with little knowledge about the dynamics of the surfboard must be very cautious if he wishes to retain stability (i.e. not falling of the board).

To be more specific the idea of the windsurfer scheme is the following. Consider the global control criterion

$$\left\| \frac{G_0 C}{1 + G_0 C} - T^d \right\|_\infty \tag{5.195}$$

where G_0 is the unknown plant, C is the controller and T^d is a desired closed-loop transfer function. We wish to achieve a minimization of this criterion by means of iterative system identification and control design.

A key ingredient in the windsurfer algorithm is the choice of reference model T^d. This model is a specification of the performance requirement and hence, in accordance with the philosophy of the approach, the performance model should be updated to cautiously increase the closed-loop bandwidth. Denote by $\hat{G}_i$ the i'th estimated model. Then $T^d_{j,i}$ is the reference model of the j'th iteration of controller refinement based on the model $\hat{G}_i$, that is, j is associated with the inner loop of Figure 5.17 and i with the outer loop. The controller $C_{j,i}$ designed using $T^d_{j,i}$ is said to robustly stabilize the model $\hat{G}_i$ if both the model and the true plant G_0 are stabilized by $C_{j,i}$. The algorithm proceeds as follows. With each of the models $\hat{G}_i$ find a sequence of robustly stabilizing controllers $C_{j,i}$ chosen such that

$$C_{j,i} = \arg\min_{K} \left\| \frac{\hat{G}_i K}{1 + \hat{G}_i K} - T^d_{j,i} \right\|_\infty \tag{5.196}$$

where the sequence of reference models $T^d_{j,i}$ generally has increasing bandwidth. The control design is performed using Internal Model Control (IMC) and the reference is of the form

$$T^d_{j,i}(s) = \left(\frac{\lambda_{j,i}}{s + \lambda_{j,i}} \right)^{n+1} \tag{5.197}$$

where $\lambda_{j,i}$ is the specified closed-loop bandwidth and n is the relative degree of the model $\hat{G}_i$. At a certain iteration $j = N$, say, a further increase in closed-loop bandwidth is not possible due to the modelling error of $\hat{G}_i$. This happens when the discrepancy between the achieved and the designed closed-loop, that is

$$\left\| \frac{G_0 C_{N,i}}{1 + G_0 C_{N,i}} - \frac{\hat{G}_i C_{N,i}}{1 + \hat{G}_i C_{N,i}} \right\|_\infty \tag{5.198}$$

is no longer small. Then a new model with improved accuracy should be found according to the control-relevant identification problem

$$\hat{G}_{i+1} = \arg\min_{\overline{G}} \left\| \frac{G_0 C_{N,i}}{1 + G_0 C_{N,i}} - \frac{\overline{G} C_{N,i}}{1 + \overline{G} C_{N,i}} \right\|_\infty \tag{5.199}$$

exactly as the problem (5.154) set up in the Schrama scheme. The solution proposed in (Lee *et al.* 1995) of this control-relevant identification problem is closely related to that used in the Schrama scheme. Again the infeasible ∞-norm problem is replaced by a 2-norm problem and the closed-loop problem is transformed to an open-loop one by introducing a certain parametrization of the system. As a part of the solution a stable factor of this parametrization is estimated using a filter derived by considering the closed-loop output error similar to Section 5.6.1.

A problem with the windsurfer approach is, that visual inspection is needed to find out when to reidentify. Also the algorithm can only handle stable plants and controllers.

5.7.5 A convergent iterative restricted complexity control design scheme

A very interesting new iterative scheme has been presented in (Hjalmarsson *et al.* 1994b). In this approach an LQG criterion parameterized directly by the controller parameters is minimized by numerical search. The gradient of the criterion is determined using signals obtained from closed-loop experiments. The controller is updated by taking steps in a descent direction; for instance in the negative gradient direction or in an approximate Gauss-Newton direction. One disadvantage of the approach, is that the gradient calculation to some extent dictate the operational conditions under which experiments are performed.

Consider a true system given by

$$y(t) = G_0(q)u(t) + v(t) \tag{5.200}$$

where $\{v(t)\}$ is a disturbance signal and the following two-degree of freedom controller

$$u(t) = C_1(q,\rho)r(t) - C_2(q,\rho)y(t) \tag{5.201}$$

where $r(t)$ is a reference signal and ρ is the parameter vector of some parametrization of the control transfer functions $C_1(q,\rho)$ and $C_2(q,\rho)$. No particular parametrization is assumed; it may be direct or indirect as a function of a system parameter vector θ by means of a model $\hat{G}(\theta)$. Signals generated in closed-loop with the above controller acting on the system will be given the argument ρ, that is $y(\rho)$ is the output of the system (5.200) controlled by (5.201). The time-argument of signals is omitted in the following.

Design criterion

Let T_d denote a desired stable transfer function from reference $r(t)$ to output $y(t)$. Define the error signal

$$\tilde{y}(\rho) = y(\rho) - T_d r = \frac{G_0 C_1(\rho)}{1 + G_0 C_2(\rho)} r - T_d r + \frac{1}{1 + G_0 C_2(\rho)} v \tag{5.202}$$

The design problem under consideration is of the type

$$\rho^* = \arg\min_{\rho} J(\rho) \tag{5.203}$$

with a quadratic criterion function of the form

$$J(\rho) = \frac{1}{2}\mathcal{E}\left\{(\tilde{y}(\rho))^2\right\} + \frac{\lambda}{2}\mathcal{E}\left\{(u(\rho))^2\right\} \tag{5.204}$$

Define the achieved closed-loop complementary sensitivity function

$$T_0(\rho) = \frac{G_0 C_1(\rho)}{1 + G_0 C_2(\rho)} \tag{5.205}$$

and the achieved sensitivity function

$$S_0(\rho) = \frac{1}{1 + G_0 C_2(\rho)} \tag{5.206}$$

Then, utilizing the independence between r and v the criterion function can be written as

$$J(\rho) = \frac{1}{2}\mathcal{E}\left\{[(T_d - T_0(\rho))r]^2 + [S_0(\rho)v]^2\right\} + \frac{\lambda}{2}\mathcal{E}\left\{[u(\rho)]^2\right\} \tag{5.207}$$

where we identify terms representing the tracking error, the noise contribution and the control penalty.

Criterion minimization

It is obvious, that the criterion function is a relatively complicated non-linear function of ρ. In addition $J(\rho)$ depends on the true system G_0 and the noise spectrum, both of which are unknown. The problem to be solved is

$$0 = J'(\rho) = \mathcal{E}\left\{\tilde{y}(\rho)\tilde{y}'(\rho)\right\} + \mathcal{E}\left\{u(\rho)u'(\rho)\right\} \tag{5.208}$$

where the prime denote differentiation with respect to the parameter vector ρ.

The gradients of signals $\tilde{y}(\rho)$ and $\tilde{u}(\rho)$ are found by performing experiments on the true plant. The gradient of $\tilde{y}(\rho)$ can be derived directly from (5.202)

$$\begin{aligned}
\tilde{y}'(\rho) &= \frac{G_0 C_1'(\rho)}{1+G_0C_2(\rho)}r - \frac{C_1(\rho)G_0^2C_2'(\rho)}{(1+G_0C_2(\rho))^2}r - \frac{G_0C_2'(\rho)}{(1+G_0C_2(\rho))^2}v \\
&= \frac{C_1'(\rho)}{C_1(\rho)}T_0(\rho)r - \frac{C_2'(\rho)}{C_1(\rho)}\left(T_0^2(\rho)r + T_0(\rho)S_0(\rho)v\right) \qquad (5.209)
\end{aligned}$$

The controller transfer functions $C_1(\rho)$ and $C_2(\rho)$ and their primed versions are all known function of ρ. This is not true for the system transfer function $S_0(\rho)$ and $T_0(\rho)$ and hence the generation of the gradient calls for experiments on the true system. Consider three experiments with reference sequences $\{r_i^j\}$, $j = 1,2,3$ and corresponding output sequences $\{y_i^j\}$, $j = 1,2,3$ where i is the iteration number of the iterative algorithm. Select

$$r_i^1 = r \tag{5.210}$$
$$r_i^2 = y_i^1 \tag{5.211}$$
$$r_i^3 = r \tag{5.212}$$

Then

$$y_i^1 = T_0(\rho_i)r + S_0(\rho_i)v_i^1 \tag{5.213}$$
$$y_i^2 = T_0^2(\rho_i)r + T_0(\rho_i)S_0(\rho_i)v_i^1 + S_0(\rho_i)v_i^2 \tag{5.214}$$
$$y_i^3 = T_0(\rho_i)r + S_0(\rho_i)v_i^3 \tag{5.215}$$

where v_i^j denote the disturbance signal at experiment j. We can now calculate exactly

$$\tilde{y}(\rho_i) = y_i^1 - T_d r \tag{5.216}$$

On noting, that

$$\tilde{y}'(\rho_i) = \frac{1}{C_1(\rho_i)}\left(C_1'(\rho_i)y_i^3 - C_2'(\rho_i)y_i^2\right) - \frac{S_0(\rho_i)}{C_1(\rho_i)}\left(C_1'(\rho_i)v_i^3 - C_2'(\rho_i)v_i^2\right) \tag{5.217}$$

it appears, that the output gradient can be estimated by

$$\hat{\tilde{y}}_i{}' = \frac{1}{C_1(\rho_i)}\left(C_1'(\rho_i)y_i^3 - C_2'(\rho_i)y_i^2\right) \tag{5.218}$$

Similarly the gradient of the input signal can be expressed as

$$u'(\rho) = \frac{C_1'(\rho)}{C_1(\rho)}C_1(\rho)S_0(\rho)r - \frac{C_2'(\rho)}{C_1(\rho)}C_1(\rho)S_0(\rho)\left(T_0(\rho)r + S_0(\rho)v\right) \tag{5.219}$$

The input signals generated in the three experiments are

$$u_i^1 = C_1(\rho_i)S_0(\rho_i)r - C_2(\rho_i)S_0(\rho_i)v_i^1 \tag{5.220}$$

$$u_i^2 = C_1(\rho_i)S_0(\rho_i)\left(T_0(\rho_i)r + S_0(\rho_i)v_i^1\right) - C_2(\rho_i)S_0(\rho_i)v_i^2 \tag{5.221}$$

$$u_i^3 = C_1(\rho_i)S_0(\rho_i)r - C_2(\rho_i)S_0(\rho_i)v_i^3 \tag{5.222}$$

It now exactly holds, that

$$u(\rho_i) = u_i^1 \tag{5.223}$$

and since

$$\begin{aligned} u'(\rho_i) &= \frac{1}{C_1(\rho_i)}\left(C_1'(\rho_i)u_i^3 - C_2'(\rho_i)u_i^2\right) \\ &\qquad - \frac{S_0(\rho_i)C_2(\rho_i)}{C_1(\rho_i)}\left(C_2'(\rho_i)v_i^2 - C_1'(\rho_i)v_i^3\right) \end{aligned} \tag{5.224}$$

the input gradient can be approximated by

$$\hat{u}_i' = \frac{1}{C_1(\rho_i)}\left(C_1'(\rho_i)u_i^3 - C_2'(\rho_i)u_i^2\right) \tag{5.225}$$

With all these signals in hand we can approximate the gradient of the criterion function as

$$\hat{J}_i' = \frac{1}{N}\sum_{t=1}^{N}\left(\tilde{y}_i(t)\hat{\tilde{y}}_i' + \lambda u_i(t)\hat{u}_i'(t)\right) \tag{5.226}$$

Iterative algorithm

1. With $C_1(\rho_i)$ and $C_2(\rho_i)$ acting on the plant, generate signals y_i^1, y_i^2, y_i^3
2. Compute $\tilde{y}_i$, $\hat{\tilde{y}}'_i$, u_i and $\hat{u}'_i$
3. Compute the next controller parameters as

$$\rho_{i+1} = \rho_i - \gamma_i R_i^{-1} \hat{J}'_i \tag{5.227}$$

   where $\{\gamma_i\}$ is a sequence of positive real number determining the step size and $\{R_i\}$ is a sequence of positive definite matrices.

4. Loop to 1 with $i := i + 1$.

The search direction R_i can be chosen in many different ways. A simple choice is $R_i = I$ which gives the negative gradient direction. (Hjalmarsson *et al.* 1994a) suggests the approximate Gauss-Newton search direction

$$R_i = \frac{1}{N} \sum_{t=1}^{N} \left(\hat{\tilde{y}}'_i \left[\hat{\tilde{y}}'_i \right]^\top + \lambda \hat{u}'_i(t) \left[\hat{u}'_i(t) \right]^\top \right) \tag{5.228}$$

(Hjalmarsson *et al.* 1994a) proves the following convergence result for the iterative algorithm.

Theorem 19 *(Hjalmarsson* et al. *1994a)*

Consider the above algorithm with $R_i \geq \delta I \ \forall i$ for some $\delta > 0$. Assume, that

$$\sum_{i=1}^{\infty} \gamma_i = \infty, \quad \sum_{i=1}^{\infty} \gamma_i^2 < \infty \tag{5.229}$$

Let the reference signal $\{r\}$ and the disturbances in each experiment $\{v_i^j\}$ be realizations of bounded mutually independent stationary stochastic processes.

Then provided, that the signals $\{y_i^j\}$; $j = 1, 2, 3$; $i = 1, 2, \ldots$ *stay bounded,*

$$\rho_i \to \{\rho : \; J'(\rho) = 0\} \quad w.p.1 \tag{5.230}$$

□

As noted in (Hjalmarsson *et al.* 1994a) this seems to be the first convergence result for an iterative control design scheme with low complexity models. The proof is based on stochastic averaging theory. There are some technicalities not mentioned here in situations where C_r is non-minimumphase or the gradients of C_2 and/or C_1 are unstable. This can be handled by introducing a properly selected all-pass filter as described in (Hjalmarsson *et al.* 1994a).

In (Hjalmarsson *et al.* 1995) it is shown, that with modelling errors the identification step taken in iterative identification and control schemes such as the Zang-scheme or using prefiltering as in Section 5.6.2 does not necessarily converge to a local minimum of the design criterion, see also (Åström and Nilsson 1994). As opposed to this, according to the above theorem, the direct minimization procedure does converge assuming bounded closed-loop signals. An inconvenience with the procedure proposed by Hjalmarsson is the fact, that three experiments must be conducted at each iteration and in the second experiment the reference signal is not free to choose.

An application of this iterative scheme to a flexible transmission benchmark system is presented in (Hjalmarsson and Gunnarsson 1995). Some considerations for slowly time-varying systems are given in (Hjalmarsson 1995).

Another related model-free approach to control design using stochastic approximation has been reported in (Spall and Cristion 1994). This paper uses a smoothed simultaneous perturbation gradient approximation technique to estimate directly an optimal controller for a minimum-variance control criterion. The control input is generated by a neural network and the controller is updated adaptively. Under some conditions on the required complexity of the controller and the smoothness of the criterion function and update of the gradient estimate asymptotic convergence to an optimal solution is proved.

5.8 Combined adaptive/iterative approaches

The rate of improvement in the iterative schemes of the closed-loop performance is limited by the length of data-collection in each iteration. As parameter estimation is based solely on data from the last iteration the amount of data should be of some size to reduce variance errors. Hence, improvement of the controller is not very frequent.

Another problem with the iterative schemes is time-varying systems. If for instance the system parameters jump inbetween iterations then identification results will be misleading. To get a faster convergence to a minimization of the global control criterion it may useful to use more flexible schemes for updating the prefilter and controller. Some ideas are discussed in this section, two of which are illustrated by simulation examples.

The basic idea is to run an on-line recursive parameter estimator using some sort of forgetting. We limit ourselves to standard recursive least squares with exponential forgetting; see (Parkum 1992) for details on recursive identification of time-varying systems. The parameter estimator considered here is of the form

$$\hat{\theta}(t) = \hat{\theta}(t-T) + \kappa(t)\left(y(t) - \varphi^\top(t)\hat{\theta}(t-T)\right) \tag{5.231}$$

$$\kappa(t) = \frac{P(t-T)\varphi(t)}{\lambda + \varphi^\top(t)P(t-T)\varphi(t)} \tag{5.232}$$

$$P(t) = \frac{1}{\lambda}\left(I - \kappa(t)\varphi^\top(t)\right)P(t-T) \tag{5.233}$$

where t is a multiple of the sampling period T, $0 < \lambda \leq 1$ is the forgetting factor and we have a linear regression model

$$y(t) = \varphi^{\top}(t)\theta + e(t) \tag{5.234}$$

Of course, the signals used in parameter estimation are filtered before entering the estimator. The control-relevant data-filter is generally a function of the system parameter estimates, that is

$$L(q) = L(q, t, \hat{\theta}(t)) \tag{5.235}$$

The use of forgetting is very important, in particular in the start-up phase. Both the controller and the prefilter used in estimation are time-varying. Hence, the estimator should discard old data based on a controller and/or prefilter, which differs significantly from the most recent one. Otherwise, the model estimate will be based on data blended together from different operational conditions.

With this parameter estimator running on-line different options on how to update the controller and prefilter can be suggested. The proposed procedures should not be seen as direct competitors to the standard iterative approaches. Using on-line estimation requires a certain amount of supervision, which is not necessary in the iterative approaches. Also validation of models and controllers is more easily done when identification is considered an off-line problem. A third less attractive feature of combined adaptive/iterative approaches is the fact, that a possible convergence analysis is even more complicated than in iterative or purely adaptive approaches. However, as will be shown, the combined methods may provide performance improvements.

5.8.1 Adaptive control and adaptive prefilter

One possibility is to use a true adaptive controller where the controller is updated at every sample based on $\hat{\theta}(t)$. An adaptive calculation of the prefilter can be done in a number of ways.

1. Directly as function of $\hat{\theta}(t)$.
2. Low-pass filter the prefilter.
3. Separate estimation of the prefilter.

We now give some general comments on these approaches.

Direct calculation of prefilter

A control-relevant prefilter will naturally be dependent on the properties of the system under control. Knowledge of the system is embodied in the parameter estimate $\hat{\theta}$. Perhaps, the most obvious way to determine the prefilter is directly as a function of the parameter estimate. For example, according to (5.87) we could choose

$$L(q,t,\hat{\theta}(t)) = \frac{\hat{H}}{1+\hat{G}(q,\hat{\theta}(t))C_2} \tag{5.236}$$

Of course, we must ascertain that the prefilter is stable. With the above prefilter this is the case if the nominal closed-loop is stable. A simulation example is presented in Section 5.8.4.

This approach gives a prefilter with time-variations depending on the variations of the parameter estimates. This can be unfortunate, especially in the start-up phase where parameter estimates may be fluctuating.

Low-pass filtering the prefilter

A low-pass filtered version of the prefilter can be generated, for instance

$$L^f(q,t) = \sigma L^f(q,t-T) + (1-\sigma)L(q,t,\hat{\theta}(t)) \tag{5.237}$$

where $0 < \sigma < 1$. This will slow down the time-variations of the prefilter.

There are two main reasons for introducing a low-pass filtering of the prefilter. The first reason is to robustify the startup phase of the adaptive controller. Control-relevant prefiltering in estimation is mainly concerned with optimizing the performance of the closed-loop system. In adaptive control the initial phase is critical with respect to divergence. Hence, an emphasis on robust stability would be advantageous in the initial phase. One way to obtain this is to start of with a "conservative" prefilter designed with robust stability in mind and then slowly adapt this prefilter towards a performance optimizing prefilter.

The second reason is to avoid fast time variations of the prefilter. Changes of the prefilter will have a slow effect on the parameter estimates. Hence, fast time variations of the prefilter are unnecessary and will most likely slow down the convergence of the parameter estimates.

Separate estimation of prefilter

In some cases it may be possible to estimate the prefilter directly from closed-loop data. For instance, the prefilter found in Section 5.6.1 is composed by the closed-loop sensitivity function and a fixed noise model. Using the indirect transfer function estimation approach described in Section 5.5 the true sensitivity function of the closed-loop system can be estimated and from this the prefilter can be generated. The advantage of this approach is, that the adaptation of the prefilter can be tuned independently from the estimation of the system parameters.

5.8.2 Adaptive control with iterative prefilter

Perhaps a safer approach is to use an adaptive controller with an iterative prefilter. The idea is to update the parameter estimate and design a new controller at each sample. However, the prefilter used in estimation should be updated iteratively, i.e. every N samples, say. This can be regarded as a kind of low-pass filtering of the prefilter and hence, the comments given for the adaptive prefilter with low-pass filtering apply here as well.

It seems intuitively more appealing to iteratively update the prefilter. With sufficiently long periods between updates, the asymptotic effect of the prefilter on the parameter estimates is more easily understood. This is in line with the asymptotic bias distribution results, that are often used in the design of the prefilter.

5.8.3 Iterative control with iterative prefilter

As a third alternative both the controller and prefilter could be updated iteratively, possibly with different updating-frequencies. Simulations indicate, that it may be advantageous to update the controller more frequently than the prefilter.

The major difference of this scheme compared to the standard iterative approaches as described in Section 5.7.1 is the fact, that the parameter estimator is working on-line and can track time-variations of the estimated plant. Also a more frequent update of the controller will give a faster response to changes in the controlled system and perhaps a faster minimization of the global control criterion.

We present a simulation example in the next section to illustrate this approach.

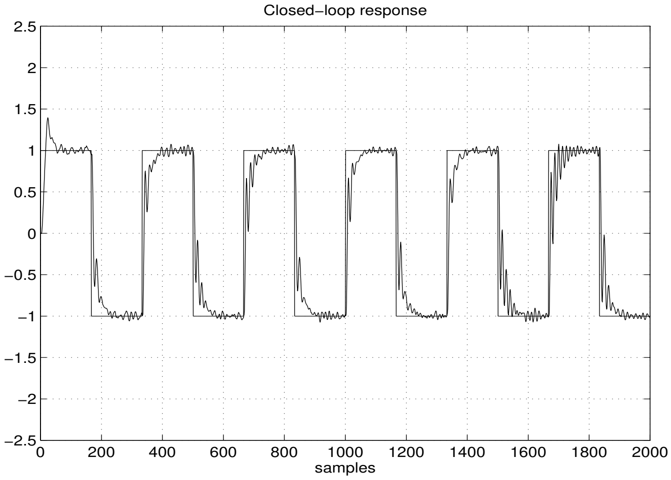

Figure 5.18: Closed-loop response, control-update every 20 samples and no prefiltering.

5.8.4 Simulation examples

Example 12 *Consider the experimental conditions and choice of controller as in Example 10. We run an on-line RLS estimator with initial covariance matrix* $P(0) = 10I$ *and a forgetting factor of* $\lambda = 0.97$*. Using an iterative update of the controller with a period of 20 samples and no prefiltering in estimation we get the closed-loop response shown in Figure 5.18. The response is very oscillating, due to an estimated model close to the stability border. We now use the equivalent prefilter for GPC (5.131) described in Section 5.6.4. The prefilter is updated every 200 samples. The closed-loop response is shown in Figure 5.19. Clearly, using control-relevant prefiltering has improved the performance. Another advantage compared to the standard iterative scheme is that poor performance in the startup phase is quickly improved, compare with Figure 5.11.* □

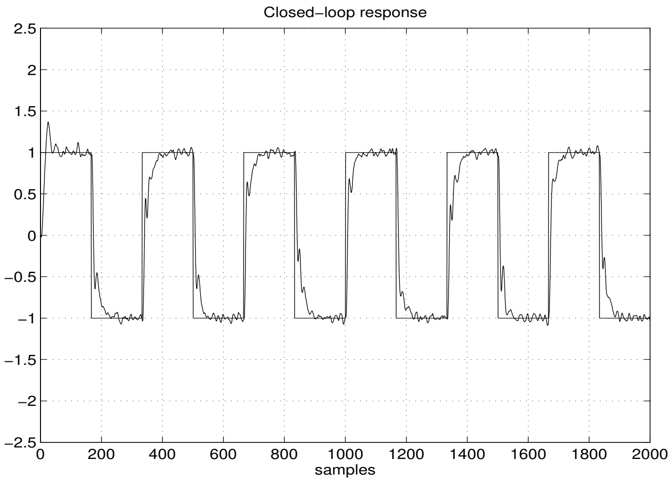

Figure 5.19: Closed-loop response, control-update every 20 samples and control-relevant GPC prefilter.

Example 13 *We continue Example 12. Now consider a true adaptive controller with an adaptive GPC prefilter update using direct calculation from the parameter estimate. We limit the experiment length to 1000 samples. The obtained response is shown in Figure 5.20. The response is very similar to that of the previous example, see Figure 5.19. Hence, in this example, there seems to be no advantage in the adaptive controller update.* □

5.9 Chapter conclusion

In this chapter different aspects on the interconnection between system identification and control design have been discussed. With low complexity models it is impossible to avoid systematic modelling errors. The system identification algorithm should take this into account. One way to affect the distribution of bias in estimation is to use prefiltering.

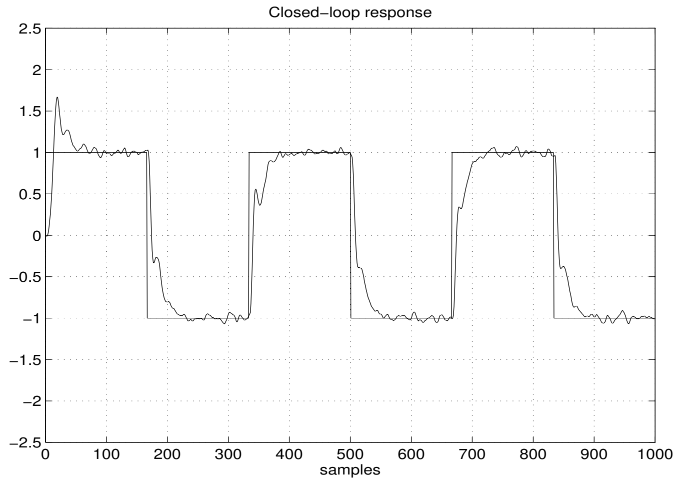

Figure 5.20: Closed-loop response, adaptive control and prefilter update.

Results on the asymptotic distribution of bias are presented, both for open-loop and closed-loop systems. These results can be used in finding control-relevant prefilters. Several strategies have been described. An important observation is, that identification should take place in closed-loop. This is intuitively appealing as the intended use of the model is to design controllers based on closed-loop data.

Recently, a number of iterative identification and control algorithms have been proposed. The idea is to minimize a global design criterion by sucessive step of local identification and control design steps. The identification problem can be seen as a way to reduce the performance degradation caused by the use of a model of the true system in the control design. By means of triangle inequalities it is possible to find control-relevant identification strategies derived directly from the control criterion. Three of the prevailing iterative algorithms using this approach have been reviewed.

As pointed out by (Hjalmarsson *et al.* 1994b) such iterative approaches have no guarantee of convergence to local minima of the global design criterion. He propose a model-free approach where the control criterion is parameterized by the controller parameters. By suitable closed-loop experiments the control criterion can be minimized directly by taking steps in descent direction. This procedure is proved to be convergent assuming bounded closed-loop signals.

Apart from a description of the above topics, a number combined adaptive/iterative approaches have been discussed. The basic idea is to use an on-line parameter estimator with forgetting. Based on the parameter estimates different strategies on the update of controller and prefilter have been suggested, some of which are illustrated by simulation examples.

To summarize, four interesting contemporary topics in the research on iterative control and identification have been illuminated in the chapter.

- *Control-relevant prefiltering:* Using existing identification techniques find based on asymptotic considerations a prefilter in estimation which enhances the relevance of the obtained model for control purposes. Typically some information of the control criterion or the obtained controller is used in the prefilter design. Update the controller, model estimate and prefilter in an iterative fashion.
- *Nonstandard control-relevant identification:* Based on the control criterion find a compatible identification criterion. The global control criterion is minimized using an iterative procedure with successive minimizations of the control and identification criterion. A difficulty with this approach is the fact that new estimation techniques must be developed for the usually non-standard identification problems that are derived from the control criterion.
- *Direct approaches:* Blend together identification and control by minimizing directly the control criterion as a function of the controller parameters without an intermediate estimation of the system model.

One such approach has been shown to have certain convergence properties.

- *Data-dependent modification of control criterion:* To emulate the global control criterion use a data-dependent modification of the local control criterion. This can be considered a bypassing of the certainty equivalence principle.

Chapter 6

Conclusion

In this thesis a number of aspects in modelling, control and identification of linear systems have been considered.

Special interest has been given to the connection between continuous-time system descriptions and their sampled counterparts. The use of δ-operator models has been shown to be superior to conventional shift-operator models at fast sampling rates. The close correspondence to continuous-time of δ-operator models has been utilized in the development of a long range prediction method which brings together discrete-time and continuous-time results. This method is subsequently used to formulate a unified framework for continuous-time and discrete-time predictive control algorithms.

Another important current research area is the development of control-relevant identification methods in face of unmodelled dynamics. It is impossible to avoid such errors when modelling physical systems. Hence, it should be taken into account in the identification algorithm in a relevant way for subsequent control design.

The following conclusions for the chapters of the thesis can be given.

Modelling of linear systems

The sampling of stochastic continuous-time linear state-space models is considered. It is found, that at fast sampling rates the conventional shift-operator is inconvenient. For that reason an alternative parametrization is introduced, which uses the so-called δ-operator. Connections between discrete-time state-space models and transfer function representations using ARMAX models have been found. Direct relationships between the parameters of q-domain and δ-domain ARMAX descriptions are established.

The following advantages of using δ-operator rather then shift-operator parametrizations can be given.

- Consistent limiting properties as the sampling period tends to zero.
- Enables unification of discrete-time and continuous-time theory.
- Easier use of physical knowledge.
- Superior numerical properties.

One important advantage of using δ-operator parametrizations is the possibility of formulating unified theoretical results for discrete-time and continuous-time models. In the limit as the sampling period tends to zero the δ-operator model converges to its continuous-time equivalent.

The use of digital computers in implementation of control algorithms makes it more natural to directly work with discrete-time estimation and control algorithms. In particular, optimal control designs can be implemented without introducing approximations.

Optimal and emulator-based prediction

Long range prediction methods intended for inclusion in predictive control algorithms have been presented. Prediction is based on ARMAX models. A number of modelling cases with varying sets of information and degree of the noise polynomial are introduced which encompass relevant situations for both continuous-time and discrete-time.

Optimal prediction, in the sense of minimum variance, using q-operator and δ-operator representations is considered. The computation of long range predictions requires solution of two coupled Diophantine-equations. Recursive algorithms to solve these equations are given. The results from the q-domain can be straightforwardly transformed to the δ-domain case.

Subsequently, a new so-called δ-domain emulator-based predictor is presented. This method is based on discrete-time models but is able to predict the output continuously. The predictor is closely related to a continuous-time prediction method and in fact converges to this in the limit as the sampling period tends to zero. It is shown, that the emulator-based predictor includes optimal prediction at the sampling instants as a special case. Hence, a link between continuous-time and discrete-time prediction has been established. This can be utilized in the development of predictive control methods.

Predictive control in the delta-domain

A unification of discrete-time and continuous-time predictive control has been obtained on two levels.

First, a δ-domain approximation of the continuous-time GPC is presented, the DCGPC. This new controller is based on a continuous-time like integral-based criterion function which is insensitive to the choice of sampling period.

Using such a criterion function is possible by using the δ-domain emulator-based predictor. The DCGPC has a well-defined limit as the sampling period tends to zero. The advantage of this approach compared to continuous-time algorithms is that no approximation is necessary in the implementation on digital computers. Also discrete-time models found by identification based on sampled data sequences can be directly applied.

Secondly, a δ-domain approximation of more conventional discrete-time predictive control methods is presented within the emulator-based framework, the DQGPC. This controller is based on a summation-based criterion and considers output predictions at the sampling instants only. With a predictor order of the emulator-based predictor equal to the number of predicted samples N_2 the output predictions are optimal. However, a computational advantage may be obtained, especially for long prediction horizons, by choosing the predictor order lower than N_2. The cost is small approximation errors.

The basic algorithms have been extended in several ways. The criterion functions have been equipped with weighting filters on the input and output. This allows a frequency dependent tuning of the control design. A new general constraint on future input in the minimization problem has been introduced. Using proper choices of this constraint and the frequency weights allows us to obtain a unified formulation which captures both the standard GPC and the new δ-domain DCGPC and DQGPC algorithms.

Finally, we have discussed state-space implementations of the basic δ-domain emulator-based controllers. It has been shown, that these can be expressed as a constant state feedback combined with a predictive observer and a feedforward from the reference signal. One advantage of the state-space formulation is the fact, that it extends easily to the multivariable case. Two different approaches to this extension have been presented.

Connecting identification and control

A number of important areas of research in control-relevant identification have been identified and illuminated.

- Control-relevant prefiltering.
- Nonstandard control-relevant identification.
- Direct approaches.
- Data-dependent criterion modification.

The concept of bias and its frequency dependent distribution is important for the construction of control-relevant prefilters in estimation. An optimal prefilter choice based on a closed-loop performance error has been found and several other control-relevant prefilters have been discussed. An important observation is, that to improve closed-loop performance identification should be based on closed-loop data.

The use of iterative algorithms in combined identification and control is motivated by the fact, that the model estimate depends on the applied controller and *vice versa*. Hence, estimation and control should not be treated independently. A truly control-relevant identification problem directly derived from the control criterion can be found by comparing the achieved and designed closed-loop. However, often this leads to nonstandard identification problems. A number of iterative algorithms have been reviewed. In one case the nonstandard identification problem can be recast to a closed-loop identification problem by proper prefiltering.

A problem with the model-based iterative schemes is that convergence is not guaranteed. A direct approach is reviewed where the criterion function is parameterized directly by the controller parameters. The parameter updates are based on gradient approximations found by closed-loop experiments. A convergence result exists for this algorithm.

The data-dependent criterion modification is a way to emulate the global design problem by introducing frequency filters based on closed-loop data. This can be considered a bypassing of the certainty equivalence principle.

Finally, new combined adaptive/iterative approaches with control-relevant prefiltering are discussed. With an on-line parameter estimator time-variations of the system may be tracked. This also enables more frequent controller and/or prefilter updates.

Appendix A

Solving Diophantine-Equations

This appendix gives proofs of the recursive solution algorithms of the coupled Diophantine-equations appearing in Chapter 3.

A.1 Optimal q-domain prediction

In this section the recursive algorithm presented in Table 3.2 will be derived. The two coupled Diophantine-equation under consideration are

$$
\begin{aligned}
q^k C_q(q) &= A_q(q) E_q^{[k]}(q) + F_q^{[k]}(q) && \text{(A.1)}\\
B_q(q) E_q^{[k]}(q) &= C_q(q) G_q^{[k]}(q) + H_q^{[k]}(q) && \text{(A.2)}\\
& && \text{(A.3)}
\end{aligned}
$$

where

$$\begin{aligned} \deg(A_q) &= n & \text{(A.4)} \\ \deg(B_q) &= m = n - d \quad \text{with } d > 0 & \text{(A.5)} \\ \deg(C_q) &= n - \Delta \quad \text{with } \Delta \in \{0,1\} & \text{(A.6)} \end{aligned}$$

A.1.1 Initialization

The solution to the Diophantine-equations is subject to the polynomial orders

$$\begin{aligned} \deg(E_q^{[j]}) &= \max\{k - \Delta, 0\} & \text{(A.7)} \\ \deg(F_q^{[j]}) &= n - 1 & \text{(A.8)} \\ \deg(G_q^{[j]}) &= \max\{k - d, 0\} & \text{(A.9)} \\ \deg(H_q^{[j]}) &= \min\{n - 1 - \Delta, m + \deg(E_q^{[j]})\} & \text{(A.10)} \end{aligned}$$

which follows from the fact that we require the transfer functions $F_q^{[k]}/A_q$ and $H_q^{[k]}/C_q$ to be strictly proper.

The initialization appears from

$$\begin{aligned} C_q &= A_q E_q^{[0]} + F_q^{[0]} & \text{(A.11)} \\ B_q E_q^{[0]} &= C_q G_q^{[0]} + H_q^{[0]} & \text{(A.12)} \\ & & \text{(A.13)} \end{aligned}$$

As both C_q and A_q are assumed to be monic it is clear, that choosing

$\Delta = 0$			$\Delta = 1$		
$E_q^{[0]}$	$=$	1	$E_q^{[0]}$	$=$	0
$F_q^{[0]}$	$=$	$C_q - A_q$	$F_q^{[0]}$	$=$	C_q
$G_q^{[0]}$	$=$	0	$G_q^{[0]}$	$=$	0
$H_q^{[0]}$	$=$	B_q	$H_q^{[0]}$	$=$	0

gives correct solutions to the Diophantine-equations and fulfills the order requirements gives above.

A.1.2 Recursion

Consider the first Diophantine-equation at step j and $j+1$

$$q^j C_q = A_q E_q^{[j]} + F_q^{[j]} \tag{A.14}$$
$$q^{j+1} C_q = A_q E_q^{[j+1]} + F_q^{[j+1]} \tag{A.15}$$

Multiply the first equation by q and subtract the two

$$0 = A_q(E_q^{[j+1]} - qE_q^{[j]}) + (F_q^{[j+1]} - qF_q^{[j]}) \tag{A.16}$$

We now choose

$$E^{[j+1]} = qE_q^{[j]} + \alpha \tag{A.17}$$

where α is a scalar. This is in agreement with the order constraint on $E_q^{[j]}$. Hence,

$$F_q^{[j+1]} = qF_q^{[j]} - \alpha A_q \tag{A.18}$$

Comparing coefficients to terms of order n and using the monicness of A_q and the fact that $\deg(F_q^{[j+1]}) = n-1$ gives

$$\alpha = f_{q,n-1}^{[j]} \tag{A.19}$$

Using a similar procedure on the second Diophantine equation we obtain

$$B_q(E_q^{[j+1]} - qE_q^{[j]}) = C_q(G_q^{[j+1]} - qG_q^{[j]}) + (H_q^{[j+1]} - qH_q^{[j]}) \tag{A.20}$$

On choosing

$$G_q^{[j+1]} = qG_q^{[j]} + \beta \tag{A.21}$$

where β is a scalar and using (A.17) we get

$$H_q^{[j+1]} = qH_q^{[j]} - \beta C_q + \alpha B_q \tag{A.22}$$

If $\Delta = 0$ then comparing coefficients to terms of order n gives

$$\beta = h_{q,n-1}^{[j]} \tag{A.23}$$

Otherwise, if $\Delta = 1$ then comparing coefficients to terms of order $n-1$ yields

$$\beta = h_{q,n-2} + \alpha b_{q,n-1} \tag{A.24}$$

Note, that if $d = n - m > 1$ then the last term in the above equation disappears.

Collecting the results presented here proves the algorithm shown in Table 3.2.

A.2 Optimal δ-domain prediction

In this section the recursive algorithm presented in Table 3.3 will be derived. The two coupled Diophantine-equation under consideration are

$$\begin{aligned} (1+\delta T)^k C_\delta(\delta) &= A_\delta(\delta) E_\delta^{[k]}(\delta) + F_\delta^{[k]}(\delta) && \text{(A.25)} \\ B_\delta(\delta) E_\delta^{[k]}(\delta) &= C_\delta(\delta) G_\delta^{[k]}(\delta) + H_\delta^{[k]}(\delta) && \text{(A.26)} \end{aligned}$$

where

$$\begin{aligned} \deg(A_\delta) &= n && \text{(A.27)} \\ \deg(B_\delta) &= m = n - d \quad \text{with } d > 0 && \text{(A.28)} \\ \deg(C_\delta) &= n - \Delta \quad \text{with } \Delta \in \{0,1\} && \text{(A.29)} \end{aligned}$$

A.2.1 Initialization

The solution to the Diophantine-equations is subject to the polynomial orders

$$\deg(E_\delta^{[j]}) = \max\{k - \Delta, 0\} \tag{A.30}$$

$$\begin{aligned} \deg(F_\delta^{[j]}) &= n-1 && \text{(A.31)} \\ \deg(G_\delta^{[j]}) &= \max\{k-d, 0\} && \text{(A.32)} \\ \deg(H_\delta^{[j]}) &= \min\{n-1-\Delta, m+\deg(E_\delta^{[j]})\} && \text{(A.33)} \end{aligned}$$

which follows from the fact that we require the transfer functions $F_\delta^{[k]}/A_\delta$ and $H_\delta^{[k]}/C_\delta$ to be strictly proper.

The initialization appears from

$$\begin{aligned} C_\delta &= A_\delta E_\delta^{[0]} + F_\delta^{[0]} && \text{(A.34)} \\ B_\delta E_\delta^{[0]} &= C_\delta G_\delta^{[0]} + H_\delta^{[0]} && \text{(A.35)} \\ & && \text{(A.36)} \end{aligned}$$

As both C_δ and A_δ are assumed to be monic it is clear, that choosing

$\Delta = 0$			$\Delta = 1$		
$E_\delta^{[0]}$	$=$	1	$E_\delta^{[0]}$	$=$	0
$F_\delta^{[0]}$	$=$	$C_\delta - A_\delta$	$F_\delta^{[0]}$	$=$	C_δ
$G_\delta^{[0]}$	$=$	0	$G_\delta^{[0]}$	$=$	0
$H_\delta^{[0]}$	$=$	B_δ	$H_\delta^{[0]}$	$=$	0

gives correct solutions to the Diophantine-equations and fulfills the order requirements gives above.

A.2.2 Recursion

Consider the first Diophantine-equation at step j and $j+1$

$$\begin{aligned} (1+\delta T)^j C_\delta &= A_\delta E_\delta^{[j]} + F_\delta^{[j]} && \text{(A.37)} \\ (1+\delta T)^{j+1} C_\delta &= A_\delta E_\delta^{[j+1]} + F_\delta^{[j+1]} && \text{(A.38)} \end{aligned}$$

Multiply the first equation by $(1+\delta T)$ and subtract the two

$$0 = A_\delta(E_\delta^{[j+1]} - (1+\delta T)E_\delta^{[j]}) + (F_\delta^{[j+1]} - (1+\delta T)F_\delta^{[j]}) \tag{A.39}$$

We now choose

$$E^{[j+1]} = (1+\delta T)E_\delta^{[j]} + \alpha \tag{A.40}$$

where α is a scalar. This is in agreement with the order constraint on $E_\delta^{[j]}$. Hence,

$$F_\delta^{[j+1]} = (1+\delta T)F_\delta^{[j]} - \alpha A_\delta \tag{A.41}$$

Comparing coefficients to terms of order n and using the monicness of A_δ and the fact that $\deg(F_\delta^{[j+1]}) = n-1$ gives

$$\alpha = T f_{\delta,n-1}^{[j]} \tag{A.42}$$

Using a similar procedure on the second Diophantine equation we obtain

$$\begin{aligned} B_\delta(E_\delta^{[j+1]} - (1+\delta T)E_\delta^{[j]}) = \\ C_\delta(G_\delta^{[j+1]} - (1+\delta T)G_\delta^{[j]}) + (H_\delta^{[j+1]} - (1+\delta T)H_\delta^{[j]}) \end{aligned} \tag{A.43}$$

On choosing

$$G_\delta^{[j+1]} = (1+\delta T)G_\delta^{[j]} + \beta \tag{A.44}$$

where β is a scalar and using (A.40) we get

$$H_\delta^{[j+1]} = (1+\delta T)H_\delta^{[j]} - \beta C_\delta + \alpha B_\delta \tag{A.45}$$

If $\Delta = 0$ then comparing coefficients to terms of order n gives

$$\beta = T h_{\delta,n-1}^{[j]} \tag{A.46}$$

Otherwise, if $\Delta = 1$ then comparing coefficients to terms of order $n-1$ yields

$$\beta = T h_{\delta,n-2} + \alpha b_{\delta,n-1} \tag{A.47}$$

Note, that if $d = n - m > 1$ then the last term in the above equation disappears.

Collecting the results presented here proves the algorithm shown in Table 3.3.

A.3 Emulator-based prediction

The two coupled Diophantine-equations which appears in the derivation of the emulator-based predictor are

$$\delta^k C_\delta(\delta) = A_\delta(\delta) E_e^{[k]}(\delta) + F_e^{[k]}(\delta) \tag{A.48}$$

$$E_e^{[k]}(\delta) B_\delta(\delta) = C_\delta(\delta) G_e^{[k]}(\delta) + H_e^{[k]}(\delta) \tag{A.49}$$

Comparison with the Diophantine-equations from the optimal q-domain predictor shows an exact algebraic correspondance between the two cases. Hence, the recursive solution of the above two Diophantine-equations follows by change of symbols from Section A.1 and will not be repeated here.

Appendix B

Some Matrix Results

B.1 A determinant formula

Theorem 20 *The following holds*

$$\det(\delta \boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C}) = \det(\delta \boldsymbol{I} - \boldsymbol{A}) + \boldsymbol{C}\,\mathrm{adj}(\delta \boldsymbol{I} - \boldsymbol{A})\boldsymbol{K} \tag{B.1}$$

□

Proof: *Consider the square matrix*

$$\boldsymbol{M} \triangleq \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{bmatrix} \tag{B.2}$$

It is well known from the matrix inversion lemma, that if $\boldsymbol{M}_{11}$ *is nonsingular then*

$$\det \boldsymbol{M} = \det \boldsymbol{M}_{11} \det(\boldsymbol{M}_{22} - \boldsymbol{M}_{21}\boldsymbol{M}_{11}^{-1}\boldsymbol{M}_{12}) \tag{B.3}$$

and if $\boldsymbol{M}_{22}$ is nonsingular then

$$\det \boldsymbol{M} = \det \boldsymbol{M}_{22} \det(\boldsymbol{M}_{11} - \boldsymbol{M}_{12}\boldsymbol{M}_{22}^{-1}\boldsymbol{M}_{21}) \tag{B.4}$$

Applying these results to the matrix

$$\boldsymbol{M} = \begin{bmatrix} \delta\boldsymbol{I} - \boldsymbol{A} & \boldsymbol{K} \\ -\boldsymbol{C} & 1 \end{bmatrix} \tag{B.5}$$

directly yields

$$\begin{aligned} \det(\delta\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{K}\boldsymbol{C}) &= \det(\delta\boldsymbol{I} - \boldsymbol{A}) \det(1 + \boldsymbol{C}(\delta\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{K}) \\ &= \det(\delta\boldsymbol{I} - \boldsymbol{A}) \left(1 + \boldsymbol{C}(\delta\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{K}\right) \\ &= \det(\delta\boldsymbol{I} - \boldsymbol{A}) + \boldsymbol{C}\operatorname{adj}(\delta\boldsymbol{I} - \boldsymbol{A})\boldsymbol{K} \end{aligned} \tag{B.6}$$

The matrix $\delta\boldsymbol{I} - \boldsymbol{A}$ is invertible except at a finite number of values of δ. Hence, the result hold from continuity considerations. ■

B.2 A result for matrix adjoints

Let

$$\boldsymbol{D} = \delta\boldsymbol{I} - \boldsymbol{A} \tag{B.7}$$

Lemma 1 *With $\boldsymbol{L}^\top$ being a row vector and $\boldsymbol{K}$ being a column vector it holds, that*

$$\boldsymbol{L}^\top \operatorname{adj}(\boldsymbol{D})\boldsymbol{K} = \boldsymbol{L}^\top \operatorname{adj}(\boldsymbol{D} + \boldsymbol{c}\boldsymbol{L}^\top)\boldsymbol{K} \tag{B.8}$$

where $\boldsymbol{c}$ is an arbitrary column vector of appropriate dimension. □

Proof: *Using the results presented in Section B.1 with*

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{c} \\ -\boldsymbol{L}^\top & 1 \end{bmatrix} \tag{B.9}$$

it follows, that

$$\det(\boldsymbol{D} + \boldsymbol{c}\boldsymbol{L}^\top) = \det(\boldsymbol{D})\left(1 + \boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c}\right) \tag{B.10}$$

By division with $\det(\boldsymbol{D})$ *we see, that (B.8) is equivalent to*

$$\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{K} = \boldsymbol{L}^\top(\boldsymbol{D} + \boldsymbol{c}\boldsymbol{L}^\top)^{-1}(1 + \boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c})\boldsymbol{K} \tag{B.11}$$

Using the matrix inversion lemma it is easily seen, that

$$\begin{aligned}(\boldsymbol{D} + \boldsymbol{c}\boldsymbol{L}^\top)^{-1} &= \boldsymbol{D}^{-1} - \boldsymbol{D}^{-1}\boldsymbol{c}(1 + \boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c})^{-1}\boldsymbol{L}^\top \boldsymbol{D}^{-1} \\ &= \boldsymbol{D}^{-1} - \boldsymbol{D}^{-1}\boldsymbol{c}\boldsymbol{L}^\top \boldsymbol{D}^{-1}(1 + \boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c})^{-1} \end{aligned} \tag{B.12}$$

Insertion into (B.11) yields

$$\begin{aligned}\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{K} &= \boldsymbol{L}^\top \boldsymbol{D}^{-1}(1 + \boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c})\boldsymbol{K} - (\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c})\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{K} \\ &= \boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{K} + (\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c})\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{K} - (\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c})\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{K} \\ &= \boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{K}\end{aligned}$$

where is has been utilized, that $\boldsymbol{L}^\top \boldsymbol{D}^{-1}\boldsymbol{c}$ *is a scalar. Throughout the derivation the involved matrix inverses exist except at a finite number of values of* δ*. Hence, the result holds from continuity considerations.* ■

A dual version of the above lemma is described below.

Lemma 2 *With* $\boldsymbol{L}^\top$ *being a row vector and* $\boldsymbol{K}$ *being a column vector it holds, that*

$$\boldsymbol{L}^\top \operatorname{adj}(\boldsymbol{D})\boldsymbol{K} = \boldsymbol{L}^\top \operatorname{adj}(\boldsymbol{D} + \boldsymbol{K}\boldsymbol{r})\boldsymbol{K} \tag{B.13}$$

where $\boldsymbol{r}$ *is an arbitrary row vector of appropriate dimension.* □

Proof: *Using the results presented in Section B.1 with*

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{D} & \boldsymbol{K} \\ -\boldsymbol{r} & 1 \end{bmatrix} \tag{B.14}$$

it follows, that

$$\det(\boldsymbol{D}+\boldsymbol{K}\boldsymbol{r}) = \det(\boldsymbol{D})\left(1+\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K}\right) \tag{B.15}$$

By division with $\det(\boldsymbol{D})$ *we see, that (B.8) is equivalent to*

$$\boldsymbol{L}^\top\boldsymbol{D}^{-1}\boldsymbol{K} = \boldsymbol{L}^\top(\boldsymbol{D}+\boldsymbol{K}\boldsymbol{r})^{-1}(1+\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K})\boldsymbol{K} \tag{B.16}$$

Using the matrix inversion lemma it is easily seen, that

$$\begin{aligned}(\boldsymbol{D}+\boldsymbol{K}\boldsymbol{r})^{-1} &= \boldsymbol{D}^{-1}-\boldsymbol{D}^{-1}\boldsymbol{K}(1+\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K})^{-1}\boldsymbol{r}\boldsymbol{D}^{-1} \\ &= \boldsymbol{D}^{-1}-\boldsymbol{D}^{-1}\boldsymbol{K}\boldsymbol{r}\boldsymbol{D}^{-1}(1+\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K})^{-1}\end{aligned} \tag{B.17}$$

Insertion into (B.16) yields

$$\begin{aligned}\boldsymbol{L}^\top\boldsymbol{D}^{-1}\boldsymbol{K} &= \boldsymbol{L}^\top\boldsymbol{D}^{-1}(1+\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K})\boldsymbol{K} - \boldsymbol{L}^\top\boldsymbol{D}^{-1}\boldsymbol{K}(\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K}) \\ &= \boldsymbol{L}^\top\boldsymbol{D}^{-1}\boldsymbol{K} + \boldsymbol{L}^\top\boldsymbol{D}^{-1}\boldsymbol{K}(\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K}) - \boldsymbol{L}^\top\boldsymbol{D}^{-1}\boldsymbol{K}(\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K}) \\ &= \boldsymbol{L}^\top\boldsymbol{D}^{-1}\boldsymbol{K}\end{aligned}$$

where is has been utilized, that $\boldsymbol{r}\boldsymbol{D}^{-1}\boldsymbol{K}$ *is a scalar. Again it has been assumed, that the matrix inverses exist.* ■

Using these to lemmae we can state the main result

Theorem 21 *It holds, that*

$$\boldsymbol{L}^\top \operatorname{adj}(\delta\boldsymbol{I}-\boldsymbol{A})\boldsymbol{K} = \boldsymbol{L}^\top \operatorname{adj}(\delta\boldsymbol{I}-\boldsymbol{A}+\boldsymbol{c}\boldsymbol{L}^\top+\boldsymbol{K}\boldsymbol{r})\boldsymbol{K} \tag{B.18}$$

where $\boldsymbol{r}$ *is an arbitrary row vector and* $\boldsymbol{c}$ *is an arbitrary column vector, both of appropriate dimension.* □

Proof: *The proof follows directly by succesive application of Lemma 1 and Lemma 2.* ■

An example illustrating a specific interpretation of the Theorem is given next.

Example 14 *Consider the simple noise-free control system configuration depicted in Figure B.1. The system equations are as follows*

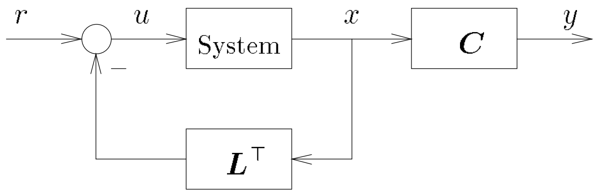

Figure B.1: A simple closed-loop configuration.

$$\delta x(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t) \tag{B.19}$$
$$y(t) = \boldsymbol{C}x(t) \tag{B.20}$$

In open-loop, i.e. with $\boldsymbol{L}^\top = 0$ *so that* $u(t) = r(t)$ *the transfer function from* r *to* y *is given by*

$$H_{yr}(\delta) = \frac{\boldsymbol{C}\,\mathrm{adj}(\delta\boldsymbol{I} - \boldsymbol{A})\boldsymbol{B}}{\det(\delta\boldsymbol{I} - \boldsymbol{A})} \tag{B.21}$$

and hence the open-loop zeros are determined as the roots of $\boldsymbol{C}\,\mathrm{adj}(\delta\boldsymbol{I} - \boldsymbol{A})\boldsymbol{B}$.

When closing the loop the system input becomes

$$u(t) = -\boldsymbol{L}^\top x(t) + r(t) \tag{B.22}$$

and the closed-loop system equations are

$$\delta x(t) = \left(\boldsymbol{A} - \boldsymbol{B}\boldsymbol{L}^\top\right) x(t) + \boldsymbol{B}r(t) \tag{B.23}$$
$$y(t) = \boldsymbol{C}x(t) \tag{B.24}$$

which gives the following transfer function from r *to* y

$$\overline{H}_{yr}(\delta) = \frac{\boldsymbol{C}\,\mathrm{adj}(\delta\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{L}^\top)\boldsymbol{B}}{\det(\delta\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{L}^\top)} \tag{B.25}$$

Hence, the closed-loop zeros are determined as the roots of $\boldsymbol{C}\,\mathrm{adj}(\delta\boldsymbol{I}-\boldsymbol{A}+\boldsymbol{B}\boldsymbol{L}^\top)\boldsymbol{B}$. *However, it is a well-known fact, that feed-back cannot move the zeros of a system. This statement is backed up by Theorem 21. When applied the theorem states, that*

$$\boldsymbol{C}\,\mathrm{adj}(\delta\boldsymbol{I}-\boldsymbol{A}+\boldsymbol{B}\boldsymbol{L}^\top)\boldsymbol{B}=\boldsymbol{C}\,\mathrm{adj}(\delta\boldsymbol{I}-\boldsymbol{A})\boldsymbol{B} \tag{B.26}$$

Hence, the zeros of the open-loop system and the zeros of the closed-loop system are determined by exactly the same polynomial, see (B.21). □

Bibliography

BEMPORAD, A., CHISCI, L., and MOSCA, E., 1994, On the stabilizing property of SIORHC, *Automatica*, **30**, 2013–2015.

BITMEAD, R. R., GEVERS, M., and WERTZ, V., 1990, *Adaptive Optimal Control: The Thinking Man's GPC*, Prentice Hall, Australia.

BLEILE, T., BOUCHER, P., and DUMUR, D., 1995, Delta-operator generalized predictive cascade control, In *ECC 95*, pp. 2857–2862, Rome, Italy.

BOUCHER, P., DUMUR, D., and NEUMANN, R., 1993, Control axis using delta-operator generalized predictive control, In *ECC 93*, volume 2, pp. 937–940, Groningen, The Netherlands.

CAINES, P. E. and BAYKAL-GÜRSOY, M., 1989, On the L^∞ consistency of L^2 estimators, *System & Control Letters*, **12**, 71–76.

CAMACHO, E. F., 1993, Constrained generalized predictive control, *IEEE Transactions on Automatic Control*, **AC-38**, 327–332.

CHISCI, L., GIARRÉ, L., and MOSCA, E., 1994, Indirect and implicit adaptive predictive control of the benchmark plant, *Automatica*, **30**, 577–584.

CHISCI, L., LOMBARDI, A., MOSCA, E., and ROSSITER, J. A., 1996, State-space approach to stabilizing stochastic predictive control, *International Journal of Control*, **65**, 619–637.

CLARKE, D. W. and MOHTADI, C., 1989, Properties of generalized predictive control, *Automatica*, **25**, 859–875.

CLARKE, D. W., MOHTADI, C., and TUFFS, P. S., 1987a, Generalized predictive control—part i: The basic algorithm, *Automatica*, **23**, 137–148.

CLARKE, D. W., MOHTADI, C., and TUFFS, P. S., 1987b, Generalized predictive control—part ii: Extensions and interpretations, *Automatica*, **23**, 149–160.

CLARKE, D. W., MOSCA, E., and SCATTOLINI, R., 1994, Robustness of an adaptive predictive controller, *IEEE Transactions on Automatic Control*, **AC-39**, 1052–1056.

CLARKE, D. W. and SCATTOLINI, R., 1991, Constrained receding-horizon predictive control, *IEE Procedings-D*, **138**, 347–354.

DE CALLAFON, R. A. and VAN DEN HOF, P. M. J., 1995a, Control relevant identification for H_∞-norm based performance specifications, In *CDC 95*, pp. 3498–3503, New Orleans, Louisiana, USA.

DE CALLAFON, R. A. and VAN DEN HOF, P. M. J., 1995b, Filtering and parametrization issues in feedback relevant identification based on fractional model representations, In *ECC 95*, pp. 441–446, Rome, Italy.

DE KEYSER, R. M. C., VAN DE VELDE, P. G. A., and DUMORTIER, F. A. G., 1988, A comparative study of self-adaptive long-range predictive control methods, *Automatica*, **24**, 149–163.

DE NICOLAO, G., SCATTOLINI, R., and SALA, G., 1996, An adaptive predictive regulator with input saturations, *Automatica*, **32**, 597–601.

DEMIRCIOĞLU, H. and CLARKE, D. W., 1992, CGPC with guaranteed stability properties, *IEE Procedings-D*, **139**, 371–380.

DEMIRCIOĞLU, H. and CLARKE, D. W., 1993, Generalised predictive control with end-point state weighting, *IEE Procedings-D*, **140**, 275–282.

DEMIRCIOĞLU, H. and GAWTHROP, P. J., 1991, Continuous-time generalized predictive control (CGPC), *Automatica*, **27**, 55–74.

DEMIRCIOĞLU, H. and GAWTHROP, P. J., 1992, Multivariable continuous-time generalized predictive control (MCGPC), *Automatica*, **28**, 697–713.

DUMUR, D. and BOUCHER, P., 1994, New predictive technics: Control axis solutions, In *3rd IEEE Conference on control applications*, pp. 1663–1668, Glasgow, Scotland, UK.

FARRELL, J. and LIVSTONE, M., 1996, Calculation of discrete-time process noise statistics for hybrid continuous/discrete-time applications, *Journal of Optimal Control Applications & Methods*, **17**, 151–155.

GARCÍA, C. E., PRETT, D. M., and MORARI, M., 1989, Model predictive control: Theory and practice—a survey, *Automatica*, **25**, 335–348.

GAWTHROP, P. J., 1987, *Continuous-Time Self-Tuning Control*, volume 1—design, Research Studies Press Ltd.

GEVERS, M., 1991, Connecting identification and robust control: a new challenge, In *Proceedings of 9th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, pp. 1–10, Budapest, Hungary.

GEVERS, M., 1993, Towards a joint design of identification and control, In Trentelman, H. L. and Willems, J. C., editors, *Essays on Control: Perspectives in the Theory and its Applications*, volume 14 of *Progress in Systems and Control Theory*, pp. 113–151. Birkhäuser.

GEVERS, M. and LI, G., 1993, *Parametrizations in Control, Estimation and Filtering Problems*, Springer-Verlag, London.

GEVERS, M. and LJUNG, L., 1986, Optimal experiment designs with respect to the intended model application, *Automatica*, **22**, 543–554.

Goodwin, G. C., Gevers, M., and Ninness, B., 1992, Quantifying the error in estimated transfer functions with application to model order selection, *IEEE Transactions on Automatic Control*, **AC-37**, 913–928.

Goodwin, G. C., Leal, R. L., Mayne, D. Q., and Middleton, R. H., 1986, Rapprochement between continuous and discrete model reference adaptive control, *Automatica*, **22**, 199–207.

Goodwin, G. C. and Salgado, M. E., 1989, A stochastic embedding approach for quantifying uncertainty in the estimation of restricted complexity models, *International Journal of Adaptive Control and Signal Processing*, **3**, 333–356.

Goodwin, G. C. and Sin, K. S., 1984, *Adaptive Filtering, Prediction and Control*, Prentice Hall, Englewood Cliffs NJ, USA.

Hagiwara, T., 1996, Analytic study on the intrinsic zeros of sampled-data systems, *IEEE Transactions on Automatic Control*, **41**, 261–263.

Hakvoort, R. G., Schrama, R. J. P., and Van den Hof, P. M. J., 1992, Approximate identification in view of LQG feedback design, In *ACC 92*, pp. 2824–2828, Chicago, Illinois, USA.

Hakvoort, R. G., Schrama, R. J. P., and Van den Hof, P. M. J., 1994, Approximate identification with closed-loop performance criterion and application to LQG feedback design, *Automatica*, **30**, 679–690.

Henson, M. A. and Seborg, D. E., 1993, Theoretical analysis of unconstrained nonlinear model predictive control, *International Journal of Control*, **58**, 1053–1080.

Hjalmarsson, H., 1995, Model free tuning of controllers: Experience with time-varying linear systems, In *ECC 95*, pp. 2869–2874, Rome, Italy.

Hjalmarsson, H., Gevers, M., De Bruyne, F., and Leblond, J., 1994a, Identification for control: closing the loop gives more accurate controllers, Technical Report 94.11, CESAME, Louvain University, Belgium.

Hjalmarsson, H. and Gunnarsson, S., 1995, Model free tuning of a robust regulator for a flexible transmission system, In *ECC 95*, pp. 1975–1980, Rome, Italy.

Hjalmarsson, H., Gunnarsson, S., and Gevers, M., 1994b, A convergent iterative restricted complexity control design scheme, In *CDC 94*, pp. 1735–1740, Lake Buena Vista, Florida, USA.

Hjalmarsson, H., Gunnarsson, S., and Gevers, M., 1995, Optimality and sub-optimality of iterative identification and control design schemes, In *ACC 95*, pp. 2559–2563, Seattle, Washington, USA.

Ishitobi, M., 1992, Conditions for stable zeros of sampled systems, *IEEE Transactions on Automatic Control*, **AC-37**, 1558–1561.

Jazwinski, A. H., 1970, *Stochastic Processes and Filtering Theory*, Academic Press.

Katayama, T., Yamamoto, T., and Wu, Y., 1995, A subspace identification of δ-operator state-space model, In *Memoirs of the Faculty of Engineering, Vol. 57, No. 2*. Kyoto University.

Kinnaert, M., 1989, Adaptive generalized predictive controller for MIMO systems, *International Journal of Control*, **50**, 161–172.

Kouvaritakis, B. and Rossiter, J. A., 1993, Multivariable stable generalised predictive control, *IEE Procedings-D*, **140**, 364–372.

Kouvaritakis, B., Rossiter, J. A., and Chang, A. O. T., 1992, Stable generalised predictive control: an algorithm with guaranteed stability, *IEE Procedings-D*, **139**, 349–362.

Kwakernaak, H. and Sivan, R., 1972, *Linear Optimal Control Systems*, Wiley-Interscience, New York, USA.

Lauritsen, M., Rostgaard, M., and Poulsen, N. K., 1994a, Emulator-based GPC in the delta-domain, Technical Report IMM-REP-1994-24, Department of Mathematical Modelling, Technical University of Denmark.

LAURITSEN, M., ROSTGAARD, M., and POULSEN, N. K., 1994b, Emulator-based prediction in the delta-domain, Technical Report IMM-REP-1994-25, Department of Mathematical Modelling, Technical University of Denmark.

LAURITSEN, M. B., 1994, Robust adaptive control, Master's thesis, IMSOR, Technical University of Denmark.

LAURITSEN, M. B., ROSTGAARD, M., and POULSEN, N. K., 1995a, Generalized predictive control in the delta-domain, In *Proceedings of the 1995 American Control Conference*, volume 5, pp. 3709–3713, Seattle, Washington, USA.

LAURITSEN, M. B., ROSTGAARD, M., and POULSEN, N. K., 1995b, Optimal prediction in the delta-domain, In *Proceedings of the 3rd European Control Conference*, volume 4, pp. 2851–2856, Rome, Italy.

LEE, W. S., ANDERSON, B. D. O., KOSUT, R. L., and MAREELS, I. M. Y., 1992, On adaptive robust control and control-relevant system identification, In *ACC 92*, pp. 2834–2841, Chicago, Illinois, USA.

LEE, W. S., ANDERSON, B. D. O., KOSUT, R. L., and MAREELS, I. M. Y., 1993, A new approach to adaptive robust control, *International Journal of Adaptive Control and Signal Processing*, **7**, 183–211.

LEE, W. S., ANDERSON, B. D. O., MAREELS, I. M. Y., and KOSUT, R. L., 1995, On some key issues in the windsurfer approach to adaptive robust control, *Automatica*, **31**, 1619–1636.

LI, G. and GEVERS, M., 1993, Comparative study of finite wordlength effects in shift and delta operator parameterizations, *IEEE Transactions on Automatic Control*, **AC-38**, 803–807.

LJUNG, L., 1985, Asymptotic variance expressions for identified black-box transfer function models, *IEEE Transactions on Automatic Control*, **AC-30**, 834–844.

LJUNG, L., 1987, *System Identification: Theory for the User*, Prentice Hall, Englewood Cliffs NJ, USA.

MAHFOUF, M. and LINKENS, D. A., 1994, Generalized predictive control (GPC) with long-range predictive identification (LRPI) for multivariable anaesthesia, *International Journal of Control*, **60**, 885–903.

MIDDLETON, R. H. and GOODWIN, G. C., 1990, *Digital Control and Estimation: A Unified Approach*, Prentice Hall, Englewood Cliffs NJ, USA.

MOHTADI, C., 1988, On the role of prefiltering in parameter estimation and control, In *Adaptive Control Strategies for Industrial Use*, volume 137, pp. 121–144. Springer Verlag.

MOSCA, E., 1995, *Optimal Predictive and Adaptive Control*, Prentice-Hall, Engelwood Cliffs NJ, USA.

MOSCA, E., ZAPPA, G., and LEMOS, J. M., 1989, Robustness of multi-predictor adaptive regulators: MUSMAR, *Automatica*, **25**, 521–529.

MOSCA, E. and ZHANG, J., 1992, Stable redesign of predictive control, *Automatica*, **28**, 1229–1233.

NEUMANN, C. P., 1993, Properties of the delta operator model of dynamic physical systems, *IEEE Transactions on Systems, Man and Cybernetics*, **23**.

ORDYS, A. W., 1993, Model-system parameter mismatch in GPC control, *International Journal of Adaptive Control and Signal Processing*, **7**.

PARKUM, J. E., 1992, *Recursive Identification of Time-varying Systems*, PhD thesis, IMSOR, Technical University of Denmark.

PARTANEN, A. G. and BITMEAD, R. R., 1993, Two stage iterative identification/controller design and direct experimental controller refinement, In *CDC 93*, pp. 2833–2838, San Antonio, Texas, USA.

PARTANEN, A. G. and BITMEAD, R. R., 1995, The application of an iterative identification and controller design to a sugar cane crushing mill, *Automatica*, **31**, 1547–1563.

RIVERA, D. E., POLLARD, J. F., and GARCÍA, C. E., 1992, Control-relevant prefiltering: A systematic design approach and case study, *IEEE Transactions on Automatic Control*, **AC-37**, 964–974.

ROBINSON, B. D. and CLARKE, D. W., 1991, Robustness effects of a prefilter in generalised predictive control, *IEE Procedings-D*, **138**, 2–8.

ROHRS, C. E., ATHANS, M., VALAVANI, L., and STEIN, G., 1984, Some design guidelines for discrete-time adaptive controllers, *Automatica*, **20**, 653–660.

ROSSITER, J. A. and KOUVARITAKIS, B., 1993, Constrained stable generalised predictive control, *IEE Procedings-D*, **140**, 243–254.

ROSSITER, J. A., KOUVARITAKIS, B., and GOSSNER, J. R., 1994, Feasibility and stability for constrained stable predictive control, In *3rd IEEE Conference on Control Applications*, pp. 1885–1890, Glasgow, Scotland, UK.

ROSTGAARD, M., 1995, *Modelling, Estimation and Control of Fast Sampled Dynamical Systems*, PhD thesis, Institute of Mathematical Modelling, Technical University of Denmark.

ROSTGAARD, M., LAURITSEN, M., and POULSEN, N. K., 1994a, Prediction using the delta operator, Technical Report IMM-REP-1994-09, Institute of Mathematical Modelling, Technical University of Denmark.

ROSTGAARD, M., LAURITSEN, M., POULSEN, N. K., and RAVN, O., 1994b, ML estimation using delta based state space models, In *SYSID 94*, volume 3, pp. 3.655–3.661, Copenhagen, Denmark.

ROSTGAARD, M., LAURITSEN, M. B., and POULSEN, N. K., 1996, A state space approach to the emulator-based GPC design, *Systems & Control Letters*, **28**, 291–301.

SCHRAMA, R. and VAN DEN HOF, P., 1993, Iterative identification and control design: A three step procedure with robustness analysis, In *ECC 93*, pp. 237–241, Groningen, The Netherlands.

SCHRAMA, R. J. P., 1992, Accurate identification for control: The necessity of an iterative scheme, *IEEE Transactions on Automatic Control*, **AC-37**, 991–994.

SCHRAMA, R. J. P. and BOSGRA, O. H., 1993, Adaptive performance enhancement by iterative identification and control design, *International Journal of Adaptive Control and Signal Processing*, **7**, 475–487.

SCHRAMA, R. J. P. and VAN DEN HOF, P. M. J., 1992, An iterative scheme for identification and control design based on coprime factorizations, In *ACC 92*, pp. 2842–2846, Chicago, Illinois, USA.

SHOOK, D. S., MOHTADI, C., and SHAH, S. L., 1991, Identification for long-range predictive control, *IEE Procedings-D*, **138**, 75–84.

SHOOK, D. S., MOHTADI, C., and SHAH, S. L., 1992, A control-relevant identification strategy for GPC, *IEEE Transactions on Automatic Control*, **AC-37**, 975–980.

SKELTON, R. E., 1989, Model error concepts in control design, *International Journal of Control*, **49**, 1725–1753.

SÖDERSTRÖM, T. and STOICA, P., 1989, *System Identification*, Prentice Hall, Hemel Hempstead, UK.

SOETERBOEK, R., 1992, *Predictive Control: A Unified Approach*, Prentice Hall, Hemel Hempstead, UK.

SPALL, J. C. and CRISTION, J. A., 1994, Nonlinear adaptive control using neural networks: Estimation with a smoothed form of simultaneous perturbation gradient approximation, *Statistica Sinica*, **4**, 1–27.

TESFAYE, A. and TOMIZUKA, M., 1995, Zeros of discretized continuous systems expressed in the Euler operator—an asymptotic analysis, *IEEE Transactions on Automatic Control*, **AC-40**, 743–747.

TESFAYE, A. and TOMIZUKA, M., 1996, Robust digital tracking with perturbation estimation via the Euler operator, *International Journal of Control*, **63**, 239–256.

Thygesen, B. G., 1993, *Modelling and Control of Systems in the Delta Domain*, PhD thesis, IMSOR, Technical University of Denmark.

Tsang, T. T. C. and Clarke, D. W., 1988, Generalised predictive control with input constraints, *IEE Procedings-D*, **135**, 451–460.

Unbehauen, H. and Rao, G. P., 1990, Continuous-time approaches to system identification—a survey, *Automatica*.

Van den Hof, P. M. J. and Schrama, R. J. P., 1993, An indirect method for transfer function estimation from closed loop data, *Automatica*, **29**, 1523–1527.

Van den Hof, P. M. J. and Schrama, R. J. P., 1995, Identification and control—closed-loop issues, *Automatica*, **31**, 1751–1770.

Van den Hof, P. M. J., Schrama, R. J. P., Bosgra, O. H., and de Callafon, R. A., 1993, Identification of normalized coprime plant factors for iterative model and controller enhancement, In *CDC 93*, pp. 2839–2844, San Antonio, Texas, USA.

Van den Hof, P. M. J., Schrama, R. J. P., de Callafon, R. A., and Bosgra, O. H., 1995, Identification of normalised coprime plant factors from closed-loop experimental data, *European Journal of Control*, **1**, 62–74.

Van der Klauw, A. C., Van Osch, J. E. F., and Van den Bosch, P. P. J., 1994, Closed-loop identification methods for LQ control design, In *SYSID 94*, pp. 3.609–3.613, Copenhagen, Denmark.

Vodă, A. and Landau, I., 1995, The joint closed loop identification and control design—theoretical issues and applications, In *ACC 95*, pp. 3551–3555, Seattle, Washington, USA.

Vodă, A. and Landau, I. D., 1994, Multi-step closed loop identification and control design procedure-applications, In *SYSID 94*, pp. 3.627–3.632, Copenhagen, Denmark.

WAHLBERG, B., 1987, *On the Identification and Approximation of Linear Systems*, PhD thesis, Department of Electrical Engineering, Linköping University, Sweden.

WAHLBERG, B. and LJUNG, L., 1986, Design variables for bias distribution in transfer function estimation, *IEEE Transactions on Automatic Control*, **AC-31**, 134–144.

WANG, W., 1994, A direct adaptive generalized predictive control algorithm for MIMO systems, *International Journal of Control*, **60**, 1371–1381.

WANG, W. and HENRIKSEN, R., 1994, A direct adaptive generalized predictive controller and some of its global convergence properties, *Journal of Modeling, Identification and Control*, **15**, 23–32.

WANG, W. and LI, G. Y., 1995, A new direct generalized predictive adaptive control algorithm, In *ECC 95*, pp. 61–64, Rome, Italy.

ZANG, Z., BITMEAD, R. R., and GEVERS, M., 1991, H_2 iterative model refinement and control robustness enhancement, In *CDC 91*, pp. 279–284, Brighton, England.

ZANG, Z., BITMEAD, R. R., and GEVERS, M., 1992, Disturbance rejection: on-line refinement of controllers by closed loop modelling, In *ACC 92*, pp. 2829–2833, Chicago, Illinois, USA.

ZANG, Z., BITMEAD, R. R., and GEVERS, M., 1995, Iterative weighted least-squares identification and weighted LQG control design, *Automatica*, **31**, 1577–1594.

ZHOU, K., DOYLE, J. C., and GLOVER, K., 1996, *Robust and Optimal Control*, Prentice Hall.

ZHU, Q. M., WARWICK, K., and DOUCE, J. L., 1991, Adaptive general predictive controller for nonlinear systems, *IEE Procedings-D*, **138**, 33–40.

ÅSTRÖM, K. J., 1970, *Introduction to Stochastic Control Theory*, Academic Press.

Åström, K. J., 1993, Matching criteria for control and identification, In *ECC 93*, pp. 248–251, Groningen, The Netherlands.

Åström, K. J., Hagander, P., and Sternby, J., 1984, Zeros of sampled systems, *Automatica*, **20**, 31–38.

Åström, K. J. and Nilsson, J., 1994, Analysis of a scheme for iterated identification and control, In *SYSID '94*, pp. 2.171–2.176, Copenhagen, Denmark.

Åström, K. J. and Wittenmark, B., 1989, *Adaptive Control*, Addison-Wesley, USA.

Åström, K. J. and Wittenmark, B., 1990, *Computer-Controlled Systems: Theory and Design*, Prentice Hall, Englewood Cliffs NJ, USA, second edition.

Ph. D. theses from IMM

1. **Larsen, Rasmus.** (1994). *Estimation of visual motion in image sequences.* *xiv* + 143 pp.
2. **Rygard, Jens Moberg.** (1994). *Design and optimization of flexible manufacturing systems.* *xiii* + 232 pp.
3. **Lassen, Niels Christian Krieger.** (1994). *Automated determination of crystal orientations from electron backscattering patterns.* *xv* + 136 pp.
4. **Melgaard, Henrik.** (1994). *Identification of physical models.* *xvii* + 246 pp.
5. **Wang, Chunyan.** (1994). *Stochastic differential equations and a biological system.* *xxii* + 153 pp.
6. **Nielsen, Allan Aasbjerg.** (1994). *Analysis of regularly and irregularly sampled spatial, multivariate, and multi-temporal data.* *xxiv* + 213 pp.
7. **Ersbøll, Annette Kjær.** (1994). *On the spatial and temporal correlations in experimentation with agricultural applications.* *xviii* + 345 pp.
8. **Møller, Dorte.** (1994). *Methods for analysis and design of heterogeneous telecommunication networks.* Volume 1-2, *xxxviii* + 282 pp., 283-569 pp.
9. **Jensen, Jens Christian.** (1995). *Teoretiske og eksperimentelle dynamiske undersøgelser af jernbanekøretøjer.* ATV Erhvervsforskerprojekt EF 435. *viii* + 174 pp.

10. **Kuhlmann, Lionel.** (1995). *On automatic visual inspection of reflective surfaces.* ATV Erhvervsforskerprojekt EF 385. Volume 1, *xviii* + 220 pp., (Volume 2, *vi* + 54 pp., fortrolig).

11. **Lazarides, Nikolaos.** (1995). *Nonlinearity in superconductivity and Josephson Junctions.* *iv* + 154 pp.

12. **Rostgaard, Morten.** (1995). *Modelling, estimation and control of fast sampled dynamical systems.* *xiv* + 348 pp.

13. **Schultz, Nette.** (1995). *Segmentation and classification of biological objects.* *xiv* + 194 pp.

14. **Jørgensen, Michael Finn.** (1995). *Nonlinear Hamiltonian systems.* *xiv* + 120 pp.

15. **Balle, Susanne M.** (1995). *Distributed-memory matrix computations.* *iii* + 101 pp.

16. **Kohl, Niklas.** (1995). *Exact methods for time constrained routing and related scheduling problems.* *xviii* + 234 pp.

17. **Rogon, Thomas.** (1995). *Porous media: Analysis, reconstruction and percolation.* *xiv* + 165 pp.

18. **Andersen, Allan Theodor.** (1995). *Modelling of packet traffic with matrix analytic methods.* *xvi* + 242 pp.

19. **Hesthaven, Jan.** (1995). *Numerical studies of unsteady coherent structures and transport in two-dimensional flows.* Risø-R-835(EN) 203 pp.

20. **Slivsgaard, Eva Charlotte.** (l995). *On the interaction between wheels and rails in railway dynamics.* *viii* + 196 pp.

21. **Hartelius, Karsten.** (1996). *Analysis of irregularly distributed points.* *xvi* + 260 pp.

22. **Hansen, Anca Daniela.** (1996). *Predictive control and identification - Applications to steering dynamics.* *xviii* + 307 pp.

23. **Sadegh, Payman.** (1996). *Experiment design and optimization in complex systems. xiv* + 162 pp.

24. **Skands, Ulrik.** (1996). *Quantitative methods for the analysis of electron microscope images. xvi* + 198 pp.

25. **Bro-Nielsen, Morten.** (1996). *Medical image registration and surgery simulation. xxvii* + 274 pp.

26. **Bendtsen, Claus.** (1996). *Parallel numerical algorithms for the solution of systems of ordinary differential equations. viii* + 79 pp.

27. **Lauritsen, Morten Bach.** (1997). *Delta-domain predictive control and identification for control. xxii* + 292 pp.