ENGINEERING DEPARTMENT
LANCASTER UNIVERSITY

Model Predictive Control Structures in the Non-Minimal State Space

by

Vasileios Exadaktylos

This thesis is submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

September 2007

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

In Memory Of My Father

Abstract

This thesis is concerned with constraint handling for systems described by a Non–Minimal State Space (NMSS) form. Such NMSS models are formulated directly from the measured input and output signals of the controlled process, without resort to the design and implementation of an observer. The thesis largely focuses on the application of Model Predictive Control (MPC) methods, a very common technique for dealing with system constraints. It is motivated by earlier research into both NMSS and MPC systems, with features of both combined in this thesis to yield improved control.

The main contribution lies in the development of new methods for constraint handling of NMSS/MPC systems that contrasts with the *ad hoc* approach previously used for NMSS design based on the Proportional–Integral–Plus (PIP) algorithm. Structural aspects of NMSS/MPC design are considered, that result from mathematical manipulation of the closed–loop block diagram or from the definition of the state space description. The properties of these structures are investigated to provide an insight on features of the proposed strategies.

More specifically, a Reference Governor scheme is utilised as a supervisory controller to account for constraints. This can lead to constraint handling in cases where a controller is already available. Furthermore, the use of an internal model is considered in the case of the 'Forward Path' NMSS/MPC controller that is shown to have improved robustness properties in comparison to the conventional 'Feedback' structure. In contrast to existing internal model approaches, this technique utilises the NMSS structure of the state vector and estimates only the elements of the state vector that are related to past values of the output. In addition, an optimal tuning technique is presented for MPC controllers. This approach allows for multiple objectives to be specified, whilst satisfying any system constraints. It is also shown that a specific NMSS/MPC structure that is proposed in this thesis,

namely the NMSS/MPC controller with an integral–of–error state, provides the designer with additional freedom when using this tuning method.

New NMSS/MPC methods are presented for both linear and non–linear systems, with the latter case being described by State Dependent Parameter (SDP) models. The development and analysis of MPC/SDP control in this thesis represents the first constraint handling control system and associated stability results for this class of non–linear models. Simulation examples are used to illustrate the advantages and potential limitations of the various control structures in comparison to existing solutions.

Acknowledgements

There are many people to whom I am indebted for their support and friendship during my time in Lancaster.

First I would like to thank my supervisor Dr. C. James Taylor for his guidance and support during my PhD studies. He has been very helpful in scientific, formality and personal matters, and the results presented in this thesis are a result of useful conversations with him. Thanks also goes to Prof. Peter Young, Dr. Arun Chotai and Dr. Liuping Wang for fruitful conversations on control and NMSS issues.

Next, I would like to thank all the people that supported me throughout the years I spent in Lancaster; too many to mention without forgetting many. Among them, special thanks goes to Kostas, Yannis and Eleni, whose friendship I shall always cherish.

I will always be grateful for the support that the General Michael Arnaoutis foundation and especially Mrs. Nathene Arnaouti has provided, both financially and personally. The Peel Trust is also acknowledged for financial support and the Faculty of Science and Technology for financially supporting my participation to the UKACC Control Conference 2006 in Glasgow.

Last, but not least, I would like to thank the close members of my family, my mother Giasemi, my brother Filippas and my grandmother Eleni for their constant support and love all these years.

Declaration

I declare that this thesis consists of original work undertaken solely by myself at Lancaster University between 2004 and 2007 and that where work by other authors is referred to, it has been properly referenced.

Vasileios Exadaktylos

September 2007

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Every physical system is subject to constraints. A car for example has a maximum acceleration (that depends on the loading of the car, the terrain it drives on, etc.) and a maximum speed (again dependent on various factors). Furthermore, safety issues impose additional constraints, such as an upper speed limit when cornering or when travelling in an urban area. The above rather simple everyday example makes apparent the need to account for various constraints when dealing with any kind of physical system.

In the same manner, it is important to account for constraints when designing control systems. When considering control systems, the presence of constraints introduces a non–linearity that complicates the analysis. In practice, to avoid this increased complexity, some control systems are designed to operate far from the constraints within specified safety bounds (e.g. a car is driven much slower than it's maximum speed allowing for safe travelling at various conditions). However, this approach is sub–optimal in the sense that it does not fully exploit the operating capability of the system that is usually achieved by operating near the constraints. Furthermore, there are many practical examples where such an approach is not acceptable. In the process control industry for example, many chemical plants

need to operate very close to or on the limits of constraints for an acceptable performance to be achieved.

Research that has previously been conducted at Lancaster University on control systems, has focused on systems described by a Non–Minimal State Space (NMSS) form and has revealed some interesting properties of such systems (see Section 1.2 for a brief description). However, there was no inherent consideration for system constraints when dealing with systems described in such a way until recently when Wang and Young (2006) proposed a Model Predictive Controller (MPC) based on a NMSS system description. In the same direction, this thesis exploits the structural properties of this controller and considers the properties of various other NMSS/MPC control systems proposed by the author. More specifically, systematic approaches of constraint handling when dealing with NMSS systems are sought by combining results from both NMSS and MPC fields.

1.2 NMSS and Proportional–Integral–Plus Control

In control design of systems described in a *minimal* state space form, a state reconstructor (that asymptotically converges to the actual system state vector) is necessary. This can be for example a deterministic state observer or the ubiquitous Kalman filter (Kalman, 1960). The introduction of such an observer results in a more complex analysis that needs to account for the observer dynamics (e.g. Żak, 2003, Chapter 3). To overcome this, Young et al. (1987) used a Non–Minimal State Space (NMSS) description of the system that is based on a state vector which consists of directly measurable system parameters (namely present and past values of the outputs and past values of the inputs). For this state space description, they developed the Proportional–Integral–Plus (PIP) control structure that can be considered as a powerful extension to the well–known Proportional–Integral–

Derivative (PID) controller.

Since then, a lot of theoretical contributions have been presented that develop and extend the NMSS/PIP control scheme. For example, Wang and Young (1988) further extended the PIP control theory; structural aspects of NMSS/PIP control have been considered by Taylor et al. (1996); Young et al. (1998) and Chotai et al. (1998) developed the PIP controller for continuous time systems described in an NMSS form; Taylor et al. (1998) evaluated a Smith Predictor to account for pure time delays in the system; while Taylor et al. (2000a) revealed the relationship between PIP and the Generalised Predictive Controller (GPC) of Clarke et al. (1987a,b).

In addition to the theoretical results, the NMSS/PIP controller has been tested in a wide range of real applications. For example, Taylor et al. (2000b) applied PIP control to a carbon dioxide enrichment process; Quanten et al. (2003) used PIP for temperature control within a car; Gu et al. (2004) used the PIP controller for an autonomous excavator; Taylor et al. (2004b) designed a PIP controller for ventilation of agricultural buildings; Van Brecht et al. (2005) controlled the 3–D distribution of air within a ventilation chamber; and Taylor et al. (2006b) applied the PIP controller to the very demanding application of a vibro–lance.

However, in the above references, constraint handling has always been an issue that was dealt with by using trial and error methods in order for the controller to avoid constraint violation as is discussed by Taylor et al. (2001). This thesis is therefore concerned with dealing with constraints using the NMSS framework; and applying techniques that have previously been developed for the NMSS/PIP controller to the proposed NMSS/MPC control structures. In this regard, the predictive control framework presented in the next section is considered.

1.3 Model Predictive Control

Model Predictive Control (MPC) is a widely used technique, especially in the chemical and process control industries (Qin and Badgwell, 1996). According to Morari (1994) MPC dates back to the 1960s, while the paper by Kwon and Pearson (1977) establishes the foundations for MPC in a form similar to the one used today. A further great step in the evolution of MPC is the Generalised Predictive Controller (GPC) of Clarke et al. (1987a,b) that initiated a whole new area of research. More recently Bemporad et al. (2002) have exploited the state feedback structure of MPC and introduced an explicit solution that makes the on–line solution of an optimisation obsolete. In this latter work, the authors divide the state space into smaller areas where a feedback controller that satisfies the constraints can be calculated. Depending on the state of the system, the appropriate feedback controller needs to be chosen degrading the optimisation problem to one of finding the appropriate feedback controller from a look–up table.

During the last decades, a lot of researchers have been working on different areas of MPC such as it's stability (for example Mayne and Michalska, 1990; Rawlings and Muske, 1993; Bemporad, 1998b; Primbs and Nevistić, 2000; Bloemen et al., 2002; Cheng and Krogh, 2001, present stability results for various MPC problem formulations) or it's optimality (e.g. Scokaert et al., 1999; Kouvaritakis et al., 2000, consider sub–optimal solutions to the MPC problem). A comprehensive review of stability and optimality of MPC is performed by Mayne et al. (2000). Various robust solutions have been presented (some approaches include those of Kothare et al., 1996; Badgwell, 1997; Magni, 2002; Fukushima and Bitmead, 2005), while a review is made by Bemporad and Morari (1999). Finally, a lot of effort has been put in reducing the on–line computational load (see for example Borrelli et al., 2001; Kouvaritakis et al., 2002; Bacic et al., 2003; Grieder et al., 2004; Imsland et al., 2005, and references therein).

The research presented in this thesis follows from the research of Wang and

Young (2006) that presented a MPC scheme using a NMSS framework (another NMSS approach to MPC can be found in Chisci and Mosca, 1994) that led to improved results compared to existing structures using a minimal state space description. Variations of the NMSS/MPC structure are considered and the NMSS system description is exploited to allow for improved simulation results when compared to existing *minimal* representations of MPC.

1.4 Academic Contribution

As already stated, the main topic of this thesis is to incorporate systematic constraint handling techniques into the NMSS framework extending the work of Wang and Young (2006); and to use experience and methodology developed for the PIP controller to the proposed MPC/NMSS control structures.

More specifically research carried out has contributed towards:

- *The importance of structure in the NMSS/MPC control scheme.* Various predictive structures are presented throughout the main body of this thesis. By exploiting the properties of NMSS models and considering approaches evaluated for PIP control, different model descriptions are considered (Chapters 3, 4 and 7), their properties are presented and comparisons among them are made. Furthermore, an alternative NMSS/MPC control structure that makes use of an internal model is considered (namely the Forward Path Controller, Chapter 6); it's properties are presented and it's performance is compared to conventional NMSS/MPC in the presence of model uncertainty.

- *Tuning of MPC Controllers.* An automatic tuning technique is presented for MPC controllers (Chapter 5). The approach is applicable to controllers based on both *minimal* and *non–minimal* state space descriptions. This is the first methodical tuning approach for MPC and allows for trade–off between different design objectives. Furthermore, it's application to different

MPC control structures is compared and it is shown that the NMSS/MPC controller introduced in Chapter 4 forms the structure of choice when using this tuning technique.

- *MPC of State Dependent Parameter (SDP) models.* Chapter 8 of the thesis considers constraint handling for the potentially highly non–linear class of SDP models. This is the first approach in this direction and can lead to better understanding of control of such systems. An iterative algorithm is proposed that utilises the predicted output to form a prediction of the system matrices. The predicted system evolution is subsequently used to form a new predicted output that is closer to the actual output of the system (based on the calculated control trajectory). In contrast to existing approaches of MPC of non–linear systems a simple stabilising algorithm is presented that is based on improved predictions and assumptions very similar to the linear case.

1.5 Thesis Outline

The rest of the thesis is organised as follows. Chapter 2 provides some necessary background mathematical concepts on system identification and optimisation. Furthermore, the notions of Receding Horizon Control (RHC) and Lyapounov stability that are used throughout the rest of the thesis are briefly reviewed.

Chapter 3 presents a reference management technique and focuses on application to existing PIP–controlled systems. Via simulation examples it is shown that a Reference Governor (RG) can provide a fast and reliable solution to constraint handling in existing control applications.

Chapter 4 introduces a NMSS/MPC controller that is based on a system description with an explicit integral–of–error state; the chapter considers it's stability, reference tracking and disturbance rejection properties. Then, Chapter 5 presents an optimisation technique for tuning MPC controllers based on multiple objec-

tives providing the necessary framework for trade–off between the objectives. It is furthermore shown that the control structure presented in Chapter 4 provides a more flexible framework compared to existing MPC realisations when using this tuning method.

Next, Chapter 6 evaluates experience from PIP control and alters the structure of the controller of Wang and Young (2006) and proposes a potentially more robust NMSS/MPC control structure, namely the Forward Path controller.

Chapter 7 considers NMSS/MPC control in the presence of modelled measured disturbances. Two structures are considered that lead to improved closed loop response when compared to the conventional NMSS/MPC approach that does not account for the disturbance model. Furthermore, a case study of an actual temperature control ventilation installation is presented and one of the proposed structures is simulated for this particular application.

Chapter 8 deals with the constraint handling problem for the highly non–linear class of SDP models. The RG approach of Chapter 3 is initially considered, while a direct application of MPC is described later in the same chapter. The evolution of the system matrices is subsequently considered to improve the state and output predictions leading to two NMSS/MPC controllers with guaranteed asymptotic stability under assumptions very similar to the case of linear systems.

Finally Chapter 9 summarises the most important results of the present research and draws future research directions.

Chapter 2

Background Concepts

This chapter introduces some concepts that are present in MPC systems. It provides the necessary background on issues that arise by presenting existing results and methods. Moreover, references are given to more advanced and detailed textbooks in each subject. However, since it is intended to be a reference chapter, formal proofs are omitted and only the essence of some ideas is highlighted.

Initially some background information on system identification techniques is given in Section 2.1. The general discrete–time difference equation that is used throughout the thesis is given and methods to identify it from open–loop simulation data are highlighted. Furthermore an introduction to estimation of the system uncertainty is given, that is used in some of the simulations in chapters that follow (in Section 6.3.2 for example).

The MPC law is derived by the minimisation of a cost function subject to constraints. Therefore, some key concepts of constrained optimisation theory are provided (Section 2.2). Additionally, a brief presentation of Receding Horizon Control (RHC), which is the key concept to applications of MPC is made (Section 2.3). Although RHC is a well understood technique, some of it's basic ideas are still presented and discussed for completeness.

The presence of constraints and the fact that MPC can handle them inherently, makes it a non–linear control technique. Therefore, in order to study it's stabil-

ity, the widely used Lyapounov theory can be employed. The basic conclusions of Lyapounov stability theory for discrete–time systems are therefore presented in Section 2.4 and reference to them is made in the main body of the thesis (e.g. in Sections 4.3.2 and 6.1.1 where the stability of different control structures is considered).

2.1 System identification

In order to develop a control algorithm, a linearised representation of the system is first required. For non–linear systems, the essential small perturbation behaviour can usually be approximated well by simple linearised transfer function models. Although different model structures have different properties in control system design (see Pearson, 2003, for a review of non–linear models in control), the present thesis considers only the following linear, q–input, p–output discrete–time system, written in difference equation form as follows,

$$\mathbf{y}_k + \mathbf{A}_1\mathbf{y}_{k-1} + \ldots + \mathbf{A}_n\mathbf{y}_{k-n} = \mathbf{B}_1\mathbf{u}_{k-1} + \mathbf{B}_2\mathbf{u}_{k-2} + \ldots + \mathbf{B}_m\mathbf{u}_{k-m} \qquad (2.1)$$

where the subscript k is a sampling index (i.e. $y_k = y(k)$); $\mathbf{y} = [y_1\ y_2\ \ldots\ y_p]^T$ and $\mathbf{u} = [u_1\ u_2\ \ldots\ u_q]^T$ are the vectors of output and input variables respectively; while $\mathbf{A}_i,\ i = 1, 2, ..., n$ and $\mathbf{B}_i,\ i = 1, 2, ..., m$ are $p \times p$ and $p \times q$ matrices of system coefficients.

For NMSS design, the model (2.1) is usually identified from measured input–output data, collected either from planned experiments (as for example in Section 7.4.2) or during the normal operation of the plant. Alternatively, they are obtained from a data–based model reduction exercise conducted on the high order simulation model (as for example in Section 4.4.2 for the IFAC '93 benchmark and in Section 6.3.4 for the ALSTOM gasifier simulation benchmark). Here, the analysis utilises the Refined and Simplified Refined Instrumental Variable (RIV/SRIV)

algorithms (Young, 1976, 1984, 1991), since these are optimal in statistical terms and often more robust to noise model specification than alternative estimation procedures (Young, 2006). Such statistical estimation analysis results in an estimate $\hat{\mathbf{p}}$ of the reduced order parameter vector $\mathbf{p}$ and an associated covariance matrix $\mathbf{P}^*$. For transfer function models, the standard errors on the parameter estimates are then computed directly from the square root of the diagonal elements of $\mathbf{P}^*$. Also, $\mathbf{P}^*$ provides an estimate of the uncertainty associated with the model parameters, which can be employed in Monte Carlo analysis to investigate the robustness of various control designs.

Finally, for a given physical system, an appropriate model structure first needs to be identified, i.e. the order of the various polynomials and the pure time delays in sampling intervals. In the latter regard, it is straightforward to introduce into equation (2.1) a time delay between the control signal and the output by assuming that the appropriate coefficients in $\mathbf{B}_i$ are zero. The two main statistical measures employed to help determine the model structure are the coefficient of determination R_T^2, based on the response error, which is a simple measure of model fit; and the Young Identification Criterion (YIC), which provides a combined measure of fit and parametric efficiency, with large negative values indicating a model which explains the output data well, without over–parametrisation (Young, 1991).

2.2 Optimisation techniques

This section provides some background concepts on optimisation and the related mathematical notions. For brevity and simplicity, formal proofs of theorems are omitted, but reference is made to publications in which they can be found. At this point it should be noted that optimisation is a huge area of ongoing research and a great number of publications exist. However, this is just a simple presentation of some results that are used specifically in MPC.

There is a huge amount of work in the area of constrained optimisation by

numerous researchers and different research groups around the world (e.g. Biegler, 1998; Borrelli et al., 2001; Chisci and Zappa, 1999; Diehl and Björnberg, 2004; Rao et al., 1998) that is not covered here. Moreover, specific attention should be paid on explicit solutions to the MPC problem (e.g. Bemporad et al., 2002; Grieder et al., 2004; Rossiter and Grieder, 2005) that make the online optimisation obsolete. However, this is out of the scope of this thesis and such methods are not presented here.

In the following, mention is made to convexity of sets and functions that is important in constrained optimisation (and is the same as the MPC problem considered throughout the thesis). Definitions and basic theorems related to convexity are presented in Appendix B.

2.2.1 The general constrained optimisation problem

The general form of a constrained optimisation problem can be described as follows:

$$\text{minimise } f(\mathbf{x}) \tag{2.2a}$$

$$\text{subject to: } g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, m \tag{2.2b}$$

$$h_j(\mathbf{x}) = 0 \quad j = 1, 2, \ldots, l \tag{2.2c}$$

$$\mathbf{x} \in \mathcal{X} \tag{2.2d}$$

The next theorem gives the conditions under which local and global minima for the problem (2.2) coincide, along with a condition for the uniqueness of a solution.

Theorem 2.2.1 (Global and unique minima (Goodwin et al., 2005)). *Consider the problem defined in* (2.2)*, where* $\mathcal{X}$ *is a nonempty convex set in* $\Re^n$*, and* $f : \mathcal{X} \to \Re$ *is convex on* $\mathcal{X}$*. If* $\mathbf{x}^* \in \mathcal{X}$ *is a local optimal solution to the problem, then* $\mathbf{x}^*$ *is a global optimal solution. Furthermore, if either* $\mathbf{x}^*$ *is a strict local minimum, or if* f *is strictly convex, then* $\mathbf{x}^*$ *is the unique global optimal solution.*

Next, the Karush–Kuhn–Tucker (KKT) necessary and sufficient conditions are presented. They provide the necessary and sufficient conditions under which a feasible[1] solution of the problem (2.2) is a local or global minimum. Combining them with Theorem 2.2.1, the conditions for a feasible solution of (2.2) to be a unique global minimum can be derived. Furthermore, the KKT conditions are employed by algorithms that solve the problem (2.2) such as the ones presented in Sections 2.2.3 and 2.2.4.

Theorem 2.2.2 (Karush–Kuhn–Tucker Necessary Conditions (Goodwin et al., 2005)). *Let $\mathcal{X}$ be a nonempty open set in $\Re^n$, and let $f : \Re^n \to \Re$, $g_i : \Re^n \to \Re$ and $h_j : \Re^n \to \Re$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, l$. Consider the problem defined in (2.2). Let $\bar{\mathbf{x}}$ be a feasible solution, and $\mathcal{I} = \{i : g_i(\bar{\mathbf{x}}) = 0\}$. Suppose that f and g_i, for $i \in \mathcal{I}$ are differentiable at $\bar{\mathbf{x}}$, that each g_i, for $i \notin \mathcal{I}$ is continuous at $\bar{\mathbf{x}}$, and that each h_j for $j = 1, 2, \ldots, l$ is continuously differentiable at $\bar{\mathbf{x}}$. Furthermore, suppose that $\nabla g_i(\bar{\mathbf{x}})^T$ and $\nabla h_i(\bar{\mathbf{x}})^T$ for $i \in \mathcal{I}$ and $j = 1, 2, \ldots, l$ are linearly independent. If $\bar{\mathbf{x}}$ is a local optimal solution, then there exist unique scalars λ_{g_i} and λ_{h_j} for $i \in \mathcal{I}$ and $j = 1, 2, \ldots, l$, such that:*

$$\nabla f(\bar{\mathbf{x}})^T + \sum_{i \in \mathcal{I}} \lambda_{g_i} \nabla g_i(\bar{\mathbf{x}})^T + \sum_{j=1}^{l} \lambda_{h_j} \nabla h_j(\bar{\mathbf{x}})^T = 0$$
$$\lambda_{g_i} \geq 0 \quad i \in \mathcal{I}$$

Furthermore, if g_i, $i \notin \mathcal{I}$ are also differentiable at $\bar{\mathbf{x}}$, then the above conditions can be written as:

$$\nabla f(\bar{\mathbf{x}})^T + \sum_{i=1}^{m} \lambda_{g_i} \nabla g_i(\bar{\mathbf{x}})^T + \sum_{j=1}^{l} \lambda_{h_j} \nabla h_j(\bar{\mathbf{x}})^T = 0$$
$$\lambda_{g_i} g(\bar{\mathbf{x}}) = 0 \quad i = 1, 2, \ldots, m$$
$$\lambda_{g_i} \geq 0 \quad i = 1, 2, \ldots, m$$

[1] Any solution that satisfies the constraints of the optimisation problem is said to be feasible.

Theorem 2.2.3 (Karush–Kuhn–Tucker Sufficient Conditions (Goodwin et al., 2005)). *Let $\mathcal{X}$ be a nonempty open set in $\Re^n$, and let $f : \Re^n \to \Re$, $g_i : \Re^n \to \Re$ and $h_j : \Re^n \to \Re$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, l$. Consider the problem defined in (2.2). Let $\bar{\mathbf{x}}$ be a feasible solution, and $\mathcal{I} = \{i : g_i(\bar{\mathbf{x}}) = 0\}$. Suppose there exist scalars $\lambda^*_{g_i}$ and $\lambda^*_{h_j}$ for $i \in \mathcal{I}$ and $j = 1, 2, \ldots, l$, such that*

$$\nabla f(\bar{\mathbf{x}})^T + \sum_{i \in \mathcal{I}} \lambda^*_{g_i} \nabla g_i(\bar{\mathbf{x}})^T + \sum_{j=1}^{l} \lambda^*_{h_j} \nabla h_j(\bar{\mathbf{x}})^T = 0 \tag{2.3}$$

Let $\mathcal{J} = \left\{ j : \lambda^*_{h_j} > 0 \right\}$ and $\mathcal{K} = \left\{ j : \lambda^*_{h_j} < 0 \right\}$. Further, suppose that f is pseudoconvex at $\bar{\mathbf{x}}$, g_i is quasiconvex at $\bar{\mathbf{x}}$ for $i \in \mathcal{I}$, h_j is quasiconvex at $\bar{\mathbf{x}}$ for $j \in \mathcal{J}$, and h_j is quasiconcave (i.e. $-h_j$ is quasiconvex) at $\bar{\mathbf{x}}$ for $j \in \mathcal{K}$. Then $\bar{\mathbf{x}}$ is a global optimal solution to problem (2.2).

Although the form (2.2) is convenient to extract valuable results for the optimisation problem, in the MPC framework it is usually formed in a way where the constraints are presented in a more compact matrix form as is shown for example in the next section (Equations (2.4b) and (2.4c)). In this regard, it would be more convenient to write the KKT conditions in a compact matrix form as follows:

$$\begin{aligned} \nabla f(\bar{\mathbf{x}})^T + \boldsymbol{\nabla g}(\bar{\mathbf{x}})^T \boldsymbol{\lambda}_i + \boldsymbol{\nabla h}(\bar{\mathbf{x}})^T \boldsymbol{\lambda}_j &= \mathbf{0} \\ \boldsymbol{\lambda}_i^T \boldsymbol{g}(\bar{\mathbf{x}}) &= \mathbf{0} \\ \boldsymbol{\lambda}_i &\geq 0 \end{aligned}$$

where $\boldsymbol{\nabla g}(\bar{\mathbf{x}})$ is the $m \times n$ Jacobian matrix whose ith row is $\nabla g_i(\bar{\mathbf{x}})$; $\boldsymbol{\nabla h}(\bar{\mathbf{x}})$ is the $l \times n$ Jacobian matrix whose ith row is $\nabla h_i(\bar{\mathbf{x}})$; $\boldsymbol{g}(\bar{\mathbf{x}})$ is a vector whose ith element is $g_i(\bar{\mathbf{x}})$; and $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}_j$ are vectors containing the Lagrange multipliers λ_i and λ_j respectively.

The above KKT conditions describe in a mathematical form the idea that there is no direction along which the cost function $f(\mathbf{x})$ at the optimal solution is de-

creasing without violating the imposed constraints. In other words, the derivative of the cost function can be described by a linear combination of the derivatives of the constraint regions at the minimising point. An interesting geometrical interpretation of the KKT conditions can be found in Papalambros and Wilde (2000).

2.2.2 The MPC optimisation problem

Although the cost function in MPC can be of any kind, it is very common for a quadratic cost function to be minimised subject to equality and inequality constraints, while an interesting consideration of the results of using a linear cost function can be found in Rao and Rawlings (2000). The general problem formulation with a quadratic cost function that is used in the rest of the thesis is:

$$\text{minimise } \{\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{p}^T\mathbf{x}\} \tag{2.4a}$$

$$\text{subject to: } \mathbf{A}_E\mathbf{x} = \mathbf{0} \tag{2.4b}$$

$$\mathbf{A}_I\mathbf{x} \leq \mathbf{0} \tag{2.4c}$$

where, $\mathbf{A}_E$ and $\mathbf{A}_I$ are matrices that represent the equality and inequality constraints and the inequality sign refers to element by element inequalities.

From Lemmas B.1.4 and B.2.3 it follows that the optimisation problem defined above is one of minimising a convex function over a convex set and according to Theorem 2.2.1 any local minimiser is a global one. Furthermore if $\mathbf{H} \succ 0$, i.e. $\mathbf{H}$ is positive definite, then from Theorem 2.2.1 it results that this global minimiser is unique.

It follows that a necessary and sufficient condition for a point $\mathbf{x}^*$ to be a unique global minimiser of (2.4) is for the KKT conditions to be satisfied, i.e. there exist Lagrange multipliers λ_i for $i \in \mathcal{I} = \{i, \mathbf{A}_I\bar{\mathbf{x}} \leq 0\}$ and λ_j for $j \in \mathcal{E} = \{i, \mathbf{A}_E\bar{\mathbf{x}} = 0\}$

such that:

$$\mathbf{H}\bar{\mathbf{x}} + \mathbf{p}^T + \mathbf{A}_I\boldsymbol{\lambda}_i + \mathbf{A}_E\boldsymbol{\lambda}_j = \mathbf{0} \tag{2.5a}$$

$$\boldsymbol{\lambda}_i \geq \mathbf{0} \tag{2.5b}$$

$$\boldsymbol{\lambda}_j^T\left(\mathbf{A}_I^T\bar{\mathbf{x}} - \mathbf{b}_I\right) = 0 \tag{2.5c}$$

where, $\boldsymbol{\lambda}_i$ and $\boldsymbol{\lambda}_j$ are vectors containing λ_i and λ_j respectively. It should be noted here that (2.5c) is added to account for the inactive inequality constraints (for more details see Goodwin et al., 2005). Furthermore, it is clear that the inequality and equality constraints of (2.4) must also be satisfied at $\mathbf{x}^*$.

This section introduced the quadratic program that will be used throughout this thesis and presented the KKT conditions for it. It is evident that it takes a simpler form than the general one and as mentioned before, a lot of efficient algorithms exist to solve this simplified problem. The underlying ideas of the two most commonly used categories of algorithms are presented in the next two sections.

2.2.3 Active set methods

In active set methods, it is assumed that an initial feasible solution exists (although this is a difficult problem on its own, especially in large optimisation problems). At this point the active constraints are identified and a decreasing direction $\mathbf{d}$ is sought such that $f(\mathbf{x}) < f(\mathbf{x}_k)$, where $\mathbf{x}_k$ is the initial feasible solution and $\mathbf{x} = \mathbf{x}_k + \mathbf{d}$. If $\mathbf{x}$ is feasible for the original problem (2.2), then it is set as an initial condition and the same procedure is repeated. If not, then a line search is made along the direction of $\mathbf{d}$ to identify the point were feasibility is lost (i.e. the point at which a previously inactive constraint becomes active). This procedure is repeated until there is no decreasing direction, i.e. $\mathbf{d} = \mathbf{0}$.

At this instant, if the KKT conditions are satisfied, the point $\mathbf{x}$ is accepted

as a global minimum and the algorithm stops. If not, then there is a direction along which some active constraint becomes inactive and the function $f(\mathbf{x})$ has a smaller value. This constraint is identified by it's associated Lagrange multiplier in the KKT conditions (it is negative). Then a descent direction away from this constraint is calculated and the algorithm proceeds on to the next iteration.

More information on the details of active set methods can be found in Fletcher (1981) and Gill et al. (1981), while the above presentation is a summary of information found mainly in Maciejowski (2002) and Goodwin et al. (2005).

An attractive feature of active set methods is that the solution lies all the time inside the feasible region. So, in cases were the algorithm needs to be terminated before the global minimiser has been found, the solution would be feasible, which is very important in real applications (feasibility can be more important than optimality in some cases (Maciejowski, 2002)). Yet, the problem of finding an initial feasible solution is hard enough for the general case (even with the quadratic program of (2.4)) and a drawback for active set methods. However, in the case of MPC, the previous solution to the problem can provide (in the absence of large disturbances) not only a feasible solution, but also one that is close to the optimal one.

An active set method is evaluated by the quadprog function (that is used in the simulations presented in this thesis) of the MATLAB® optimisation toolbox in 'Medium–Scale' mode (i.e. in most cases considered in this thesis).

2.2.4 Interior point methods

There are a lot of different variations of interior point methods with their main characteristic being that, in contrast to the active set methods, they don't search for the optimal solution at the boundary of the constraint region. On the contrary they start from an arbitrary point and create a path (usually not geometrically intuitive) to the optimal solution. A useful property of these algorithms is the fact

that a lot of them don't necessarily require a feasible initial point (making the problem easier to solve, although a good initial guess can drastically improve the convergence of the algorithm (Melman and Polyak, 1996)). However, the majority of them will not provide a feasible solution until the end of the optimisation procedure, which can cause problems in large scale problems where a feasible solution is required even if the algorithm has not terminated (as already mentioned in the description of the Active Set methods in Section 2.2.3).

In a bit more detail, the idea underlying the primal–dual interior point methods is presented here. In this class of algorithms, a *barrier* function is introduced that is dependent on the constraint region of the initial optimisation problem and the following unconstrained problem is introduced:

$$\text{minimise } \{f(\mathbf{x}) - f_{barrier}(\mu, g_i(\mathbf{x}), h_j(\mathbf{x}))\} \tag{2.6}$$

where $f_{barrier}(\cdot)$ is a function that that monotonically decreases while $\mu \to \infty$; and $f(\cdot)$, $g_i(\cdot)$, $i = 1, \ldots, m$ and $h_j(\cdot)$, $j = 1, \ldots, l$ are defined in (2.2). Both (2.6) and it's dual are subsequently solved sequentially for increasing values of μ. It can be shown that as μ increases, the solutions of the primal and dual problems come closer and eventually converge to the solution of the initial constrained problem (2.2).

There are a lot of variations and algorithms for interior point methods in the literature. A collection of some, along with a discussion on some issues that arise can be found in Anstreicher and Freund (1996). Some intuitively nice guidelines are also given by Maciejowski (2002) and Goodwin et al. (2005), while Rao et al. (1998) present an optimisation algorithm that exposes the structure of the MPC problem for improved speed.

The quadprog function that is used in the simulations of this thesis uses an interior point method to solve 'Large–Scale' optimisation problems.

2.3 Receding Horizon Control (RHC)

As already mentioned the control law to be applied to the system results from an optimisation problem that can take the form of (2.4). To introduce the notions of prediction and control horizon, the following optimisation problem is defined:

$$\text{minimise} \sum_{i=1}^{N_p} \mathbf{x}_{k+i}^T \mathbf{Q} \mathbf{x}_{k+i} + \sum_{i=0}^{N_c-1} \mathbf{u}_{k+i}^T \mathbf{R} \mathbf{u}_{k+i} \tag{2.7a}$$

$$\text{subject to: } \mathbf{x}_{k+i} = f(\mathbf{x}_{k+i-1}, \mathbf{u}_{k+i-1}) \tag{2.7b}$$

$$\mathbf{x}_{k+i} \in \mathcal{X} \quad i = 1, \ldots, N_p \tag{2.7c}$$

$$\mathbf{u}_{k+i} \in \mathcal{U} \quad i = 0, \ldots, N_c - 1 \tag{2.7d}$$

where the minimisation is performed over the control input sequence $\{\mathbf{u}_{k+i|k}\}$, $i = 0, \ldots, N_c - 1$, where the subscript $|k$ defines the sampling instant at which the prediction is based on and $\mathcal{X}$ and $\mathcal{U}$ represent constraint sets on the state and inputs of the system respectively. This is in the form of the well–known Linear Quadratic Regulator (Mosca, 1995) with the addition of constraints, but instead of taking into account the system evolution $\mathbf{x}_{k+i} = f(\mathbf{x}_{k+i-1}, \mathbf{u}_{k+i-1})$ for $i = 1, \ldots, \infty$, only N_p steps ahead are considered, and N_p is called the prediction horizon. In the same manner N_c future values of the control signal are sought, and N_c is called the control horizon. Details on how to bring the problem (2.7) in the form of (2.4) are given in later chapters for the specific cost function that is used in each MPC formulation of the problem.

However, something unexpected can happen before the end of this horizon (when a new control sequence would be calculated). In this case the predicted system evolution differs from the actual one. To avoid this kind of mismatch, (2.7) is solved at every sampling instant and the receding horizon strategy is evaluated.

According to the Receding Horizon Control (RHC) strategy, only the first element of the computed control sequence is applied while the rest are discarded

and the optimisation problem is solved again. In more detail, the control steps are described in the following.

1. At time instant k the problem (2.7) is solved and the optimal sequence $\{\mathbf{u}_{k|k}, \ldots, \mathbf{u}_{k+N_c-1|k}\}$ is calculated.

2. The first first value $\mathbf{u}_k$ of the control sequence is applied to the system and the rest is discarded.

3. A new measurement of the system state $\mathbf{x}_{k+1}$ is made at the following sampling instant $k+1$.

4. Based on the new measurement the optimisation problem (2.7) is solved again and a new control sequence $\{\mathbf{u}_{k+1|k+1}, \ldots, \mathbf{u}_{k+N_c|k+1}\}$ is calculated.

A graphical representation of the RHC strategy is given in Figure 2.1. The prediction and control horizons are shown, while it becomes apparent that they slide along to the future, at each sampling instant.

It is clear that in the disturbance free case and when there is no model mismatch, the difference in the optimal control sequences reduces by increasing the control and prediction horizons and eventually it converges to the optimal solution of the infinite horizon control problem (in the absence of constraints).

The receding horizon control strategy allows for the controller to take into account new information about the system's environment as they become apparent. Getting closer to system constraints for example may result in the controller producing a more 'conservative' control sequence that will allow constraint satisfaction that would not have been the case at a previous sampling instant were the system was far from the constraints. Furthermore, knowledge of a particular modelled disturbance can give the controller adequate information to react before the disturbance becomes apparent in the measurements as is shown in Chapter 7. Such an example can be the surrounding temperature of a controlled environment that will affect it's temperature several samples into the future.

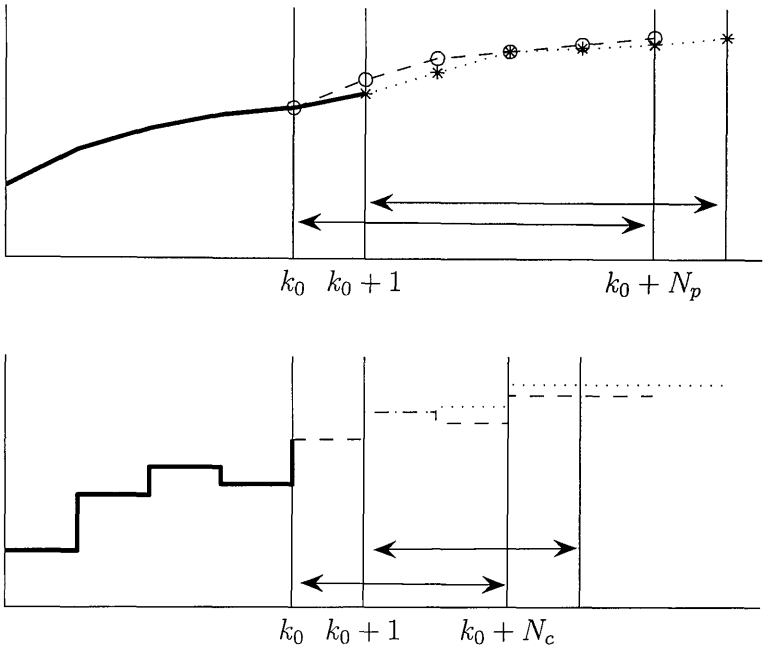

Figure 2.1: The Receding Horizon Control strategy. It is evident that a mismatch between the predictions of two consecutive sampling instant may occur.

2.4 Lyapounov theory

A very general definition of stability is the one of Bounded–Input Bounded–Output stability as defined below.

Definition 2.4.1 (Bounded-Input Bounded-Output Stability (Marlin, 1995)). *A system is stable if all output variables are bounded when all input variables are bounded*

However, the above definition is quite general and although it provides a useful insight on the notion of stability, it is not very helpful when it comes to formally addressing this issue. In this regard, the more widely used Lyapounov stability is considered below, according to which an equilibrium point is stable if a small perturbation of the state or input results in a *continuous* perturbation of the subsequent state and input trajectories. More formally, it is defined as:

Definition 2.4.2 (Lyapounov Stability (Letov, 1961)). *Let $\mathcal{S}$ be a set in $\Re^n$ that contains the origin. Let $f : \Re^{n+m} \rightarrow \Re^n$ be such that $f(\mathcal{S}) \subset \mathcal{S}$. Suppose that the*

system

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \tag{2.8}$$

with $\mathbf{x}_k \in \Re^n$, *has an equilibrium point at* $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$, *that is* $f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \bar{\mathbf{x}}$. *Let* $\mathbf{x}_{k_0} \in \mathcal{S}$ *and* $\{\mathbf{x}_{k_0+i}\} \in \mathcal{S}$, *for* $i \geq 1$, *be the resulting sequence satisfying* (2.8) *for a control sequence* $\mathbf{u}_{k_0+i}$, $i \geq 1$. *The equilibrium point is stable in* $\mathcal{S}$ *if for any* $\epsilon > 0$, *there exists* $\delta(\epsilon) > 0$ *such that*

$$\mathbf{x}_{k_0} \in \mathcal{S} \text{ and } \| \left[(\mathbf{x}_{k_0} - \bar{\mathbf{x}})^T, (\mathbf{u}_{k_0} - \bar{\mathbf{u}})^T \right]^T \| < \delta \Rightarrow$$
$$\| \left[(\mathbf{x}_{k_0+i} - \bar{\mathbf{x}})^T, (\mathbf{u}_{k_0+i} - \bar{\mathbf{u}})^T \right]^T \| < \epsilon, \ \forall i \geq 0$$

Furthermore, if in addition $\| \left[(\mathbf{x}_{k_0+i} - \bar{\mathbf{x}})^T, (\mathbf{u}_{k_0+i} - \bar{\mathbf{u}})^T \right]^T \| \to 0$ *as* $i \to \infty$ *then it is asymptotically stable.*

Since it is the most widely used stability definition and has been extensively used in the literature, in the rest of the thesis only Lyapounov stability is considered. The notion of Lyapounov stability of an equilibrium point requires the system to remain *near* that point after a disturbance in the state or the input of the system, while asymptotic stability requires for the equilibrium point to be reached eventually. The most common way of showing that a point is Lyapounov stable in nonlinear control theory is by means of Lyapounov's theorem that is presented below.

Lyapounov's Theorem. (Maciejowski, 2002). If there is a function $V(\mathbf{x}, \mathbf{u})$ which is positive definite, with $V(\mathbf{x}, \mathbf{u}) = 0$ only if $(\mathbf{x}, \mathbf{u}) = (\mathbf{0}, \mathbf{0})$, and has the property that

$$\| [\mathbf{x}_1, \mathbf{u}_1] \| > \| [\mathbf{x}_2, \mathbf{u}_2] \| \Rightarrow V(\mathbf{x}_1, \mathbf{u}_1) \geq V(\mathbf{x}_2, \mathbf{u}_2)$$

and if, along any trajectory of the system $\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k)$ in some neighbourhood

of $(\mathbf{0}, \mathbf{0})$ the property

$$V(\mathbf{x}_{k+1}, \mathbf{u}_{k+1}) \leq V(\mathbf{x}_k, \mathbf{u}_k)$$

holds, then $(\mathbf{0}, \mathbf{0})$ is a stable equilibrium point. If, in addition, $V(\mathbf{x}_k, \mathbf{u}_k) \rightarrow 0$ as $k \rightarrow \infty$ then it is asymptotically stable. Such a function is called a Lyapounov function

Lyapounov's theorem states the conditions for stability and asymptotic stability of the origin. Yet, with a change in the system coordinates any equilibrium can be moved to the origin. Therefore, the Lyapounov theorem can be applied to prove stability of any equilibrium point.

A stability proof for the general MPC scheme can be found in Appendix C that poses the basis for stability proofs throughout this thesis.

2.5 Conclusions

This chapter reviewed background concepts that are used later in this thesis. The identification procedure that is later used to obtain the model used in the predictions has been presented, along with concepts of constrained optimisation techniques. Furthermore, the Receding Horizon technique that is used in the MPC strategies presented throughout the thesis was briefly presented. Finally, concepts of Lyapounov stability were presented since this is the main tool for stability analysis.

Chapter 3

Reference governor over PIP control

As already mentioned, a common control structure for systems described in the NMSS is Proportional–Integral–Plus (PIP) control, that can be considered a powerful extension of the well known Proportional–Integral–Derivative (PID) control scheme. In the cases however where constraint handling is an issue, PIP (and PID) control is designed off–line in an *ad–hoc* manner to avoid constraint violation (e.g. Taylor et al., 2001; Taylor and Shaban, 2006, adjust the LQ weights using off–line simulations).

Two common techniques to deal with constraint handling when a controller is already available are the Closed Loop Paradigm (CLP) (e.g. Rossiter, 2003) and the Reference Governor (RG) (e.g. Bemporad et al., 1997). In both, an optimal supervisory control scheme is applied to the already controlled system to ensure constraint satisfaction. Although it can be proved that both control schemes are equivalent (Rossiter, 2003) with the only difference being in the way the degrees of freedom (dof) are parametrised, they have different properties regarding the conditioning of the resulting optimisation problem and the insight they provide as to the effect of the constraints to the closed loop system.

This chapter considers the problem formulation as in that of the RG, while

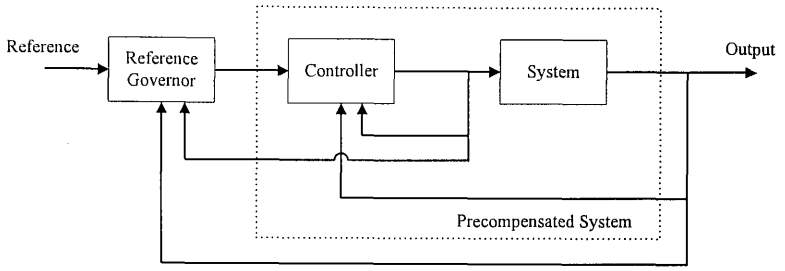

Figure 3.1: The Reference Governor Control Scheme

parametrising the dof as perturbations around the reference signal (even though this is more common in the CLP approach, it is chosen here due to it's simplicity and the insight it can provide on the effect of the constraints on the closed loop system). The underlying idea of RG techniques is that as long as a stable controller exists, there can be a reference sequence that can lead to constraint satisfaction whilst preserving the closed loop system properties for this artificial reference (or for the actual one when operating far form the constraints). It can also be considered as a nonlinear reference filter that alters the reference signal when it would cause constraint violation if it was directly applied to the system. A block diagram of the RG control scheme is presented in Figure 3.1.

The approach presented in this chapter is a first attempt for systematic constraint handling within the NMSS. It can easily be applied to systems were a PIP controller is already available and improve the performance of the overall control scheme (as is shown by a simulation example in Section 3.4). The same technique is later evaluated for an initial approach to constraint handling in the more demanding context of the non–linear class of State Dependent Parameter (SDP) models as described in Chapter 8. Reference will be made here for the basic notions of the RG scheme and formulation of the control problem.

Next, Section 3.1 presents the formulation of a PIP controller in the absence of constraints, while Section 3.2 introduces the RG. This is followed by Section 3.3 that briefly consi ers the stability and reference tracking properties of the RG scheme and Section 3.4 that compares by simulation a closed loop system with

and without a RG. Finally, Section 3.5 concludes the present chapter.

3.1 System description and closed loop control

Consider the q–input p–output system described by the difference equation (2.1). As described in Young et al. (1987), the system described by (2.1) can take the following NMSS form:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{D}\mathbf{r}_k \tag{3.1a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k \tag{3.1b}$$

where $\mathbf{r} = [r_1\ r_2\ \dots\ r_p]^T$ is the reference level vector and the $(n+1)p+m(q-1)\times 1$ state vector $\mathbf{x}$ is defined as:

$$\mathbf{x}_k = \begin{bmatrix}\mathbf{y}_k^T & \mathbf{y}_{k-1}^T & \cdots & \mathbf{y}_{k-n+1}^T & \mathbf{u}_{k-1}^T & \mathbf{u}_{k-2}^T & \cdots & \mathbf{u}_{k-m+1}^T & \mathbf{z}_k^T\end{bmatrix}^T \tag{3.2}$$

in which $\mathbf{z}_k = \frac{1}{1-z^{-1}}(\mathbf{r}_k - \mathbf{y}_k)$ is the *integral–of–error* state variable that is used to ensure type one servomechanism performance (e.g. Taylor et al., 2000a). The system matrices in (3.1) are:

$$\mathbf{A} = \begin{bmatrix}
-\mathbf{A}_1 & \cdots & -\mathbf{A}_{n-1} & -\mathbf{A}_n & \mathbf{B}_2 & \cdots & \mathbf{B}_{m-1} & \mathbf{B}_m & \mathbf{0}_p \\
\mathbf{I}_p & \cdots & \mathbf{0}_p & \mathbf{0}_p & \mathbf{0}_{pq} & \cdots & \mathbf{0}_{pq} & \mathbf{0}_{pq} & \mathbf{0}_p \\
\vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{0}_p & \cdots & \mathbf{I}_p & \mathbf{0}_p & \mathbf{0}_{pq} & \cdots & \mathbf{0}_{pq} & \mathbf{0}_{pq} & \mathbf{0}_p \\
\mathbf{0}_{qp} & \cdots & \mathbf{0}_{qp} & \mathbf{0}_{qp} & \mathbf{0}_q & \cdots & \mathbf{0}_q & \mathbf{0}_q & \mathbf{0}_{qp} \\
\mathbf{0}_{qp} & \cdots & \mathbf{0}_{qp} & \mathbf{0}_{qp} & \mathbf{I}_q & \cdots & \mathbf{0}_q & \mathbf{0}_q & \mathbf{0}_{qp} \\
\vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{0}_{qp} & \cdots & \mathbf{0}_{qp} & \mathbf{0}_{qp} & \mathbf{0}_q & \cdots & \mathbf{I}_q & \mathbf{0}_q & \mathbf{0}_{qp} \\
\mathbf{A}_1 & \cdots & \mathbf{A}_{n-1} & \mathbf{A}_n & -\mathbf{B}_2 & \cdots & -\mathbf{B}_{m-1} & -\mathbf{B}_m & \mathbf{I}_p
\end{bmatrix} \tag{3.3a}$$

$$\mathbf{B} = \left[\mathbf{B}_1^T \;\; \mathbf{0}_{pq}^T \;\; \cdots \;\; \mathbf{0}_{pq}^T \;\; \mathbf{I}_q \;\; \mathbf{0}_q \;\; \cdots \;\; \mathbf{0}_q \;\; -\mathbf{B}_1^T\right]^T \tag{3.3b}$$

$$\mathbf{D} = \left[\mathbf{0}_p \;\; \mathbf{0}_p \;\; \cdots \;\; \mathbf{0}_p \;\; \mathbf{0}_{qp}^T \;\; \mathbf{0}_{qp}^T \;\; \cdots \;\; \mathbf{0}_{qp}^T \;\; \mathbf{I}_p\right]^T \tag{3.3c}$$

$$\mathbf{C} = \left[\mathbf{I}_p \;\; \cdots \mathbf{0}_p \;\; \mathbf{0}_p \;\; \mathbf{0}_{pq} \;\; \cdots \;\; \mathbf{0}_{pq} \;\; \mathbf{0}_{pq} \;\; \mathbf{0}_p\right] \tag{3.3d}$$

where $\mathbf{I}_p$ and $\mathbf{0}_p$ are the $p \times p$ identity and zero matrices respectively, while $\mathbf{0}_{pq}$ is a $p \times q$ matrix of zeros.

The above system representation has been extensively studied and numerous applications exist that evaluate the PIP control methodology (e.g. Lees et al., 1996; Taylor et al., 2000b; Quanten et al., 2003; Gu et al., 2004). As in any control structure there are various control design methods for PIP controllers (such as pole assignment or LQ optimal design). However after designing the controller to give the system the desired properties (i.e. rise and settling time, overshoot, etc.[1]), it takes the following state feedback form:

$$\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k \tag{3.4}$$

where $\mathbf{K} = [\mathbf{K}_1 \;\; \mathbf{K}_2 \;\; \cdots \;\; \mathbf{K}_n \;\; \mathbf{K}_{n+1} \;\; \cdots \;\; \mathbf{K}_{n+m-1} \;\; \mathbf{K}_{n+m}]$ is the gain matrix. The controlled system can now be described by the free response state space model defined by:

$$\mathbf{x}_k = \tilde{\mathbf{A}}\mathbf{x}_{k-1} + \mathbf{D}\mathbf{w}_k \tag{3.5a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k \tag{3.5b}$$

in which $\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{B}\mathbf{K}$, while the slack reference vector signal $\mathbf{w}$ that is defined in the next section is used instead of the actual one. At this point $\mathbf{w}_k = \mathbf{r}_k$ is considered and it is clear that the system (3.5) is the one of (3.1) after the

[1]The controller properties hold in the case where constraints are not present or the controller operates away from the constraints. In the presence of input constraints for example, the control signal might saturate and as a consequence the system response differs from the designed one.

application of the control law (3.4).

3.2 Control Description

As mentioned before, the underlying idea of the RG is to predict the system evolution, detect any constraint violations and produce a slack reference sequence that, if applied to the system, results in constraint satisfaction. This slack reference should converge to the actual one so that there is offset free tracking of the actual reference. Although there are numerous options on how to parametrise this slack reference sequence (e.g. Bemporad et al., 1997; Bemporad, 1998a; Oh and Agrawal, 2005), a perturbation around the actual one is used in the following:

$$\mathbf{w}_k = \mathbf{r}_k + \boldsymbol{\beta}_k \tag{3.6}$$

where $\mathbf{w}_k$ is the artificial reference that is applied to the system and the perturbation vector has the form $\boldsymbol{\beta} = [\beta_1 \quad \dots \quad \beta_p]^T$. As is later shown by the simulation example (Figure 3.5), the above parametrisation of $\mathbf{w}$ can also provide a measure of the effect of the constraints on the closed loop system.

Using the parametrisation (3.6), it is sensible to try and minimise the perturbations around the reference while maintaining the system within the region defined by the constraints. In this regard, the following constrained optimisation problem is solved at each sampling instant:

$$\min_{\boldsymbol{\beta}_1,\dots,\boldsymbol{\beta}_{N_p}} \quad \sum_{i=1}^{N_p} \boldsymbol{\beta}_i^T \boldsymbol{\beta}_i \tag{3.7a}$$

$$\text{subject to:} \begin{cases} \underline{\mathbf{u}} \leq \mathbf{u}_{k+i} \leq \overline{\mathbf{u}} \\ \underline{\Delta\mathbf{u}} \leq \Delta\mathbf{u}_{k+i} \leq \overline{\Delta\mathbf{u}} \\ \underline{\mathbf{y}} \leq \mathbf{y}_{k+i+1} \leq \overline{\mathbf{y}} \end{cases} \qquad i = 0, \dots, N_p - 1 \tag{3.7b}$$

where $\mathbf{u}_k$ is given by (3.4); $\Delta\mathbf{u}_{k+i} = \mathbf{u}_{k+i} - \mathbf{u}_{k+i-1}$ is the vector of control increments at each sampling instant; the system evolution is that of (3.5); $\underline{\ }$ and $\bar{\ }$ refer to minimum and maximum allowed values for the system variables; and the inequalities in the constraints are element by element inequalities. It is clear that in the absence of constraints, the minimising $\boldsymbol{\beta}$-sequence is zero and the reference remains unchanged. Yet, when the actual reference results in constraint violation by the controlled system within the prediction horizon, the predictive control law intervenes and changes the reference input so that the constraints are satisfied.

For commercial optimisation tools to be used, the above constraint optimisation problem needs to be cast into a more compact matrix form. In this regard, the optimisation parameters are formed into a single vector as:

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T & \boldsymbol{\beta}_2^T & \dots & \boldsymbol{\beta}_{N_p}^T \end{bmatrix}^T$$

and the following predicted state, future control input, future control input increment, future reference and predicted output vectors are defined:

$$\begin{aligned}
\mathbf{X} &= \begin{bmatrix} \mathbf{x}_{k+1}^T & \mathbf{x}_{k+2}^T & \dots & \mathbf{x}_{k+N_p}^T \end{bmatrix}^T \\
\mathbf{U} &= \begin{bmatrix} \mathbf{u}_k^T & \mathbf{u}_{k+1}^T & \dots & \mathbf{u}_{k+N_p-1}^T \end{bmatrix}^T \\
\Delta\mathbf{U} &= \begin{bmatrix} \Delta\mathbf{u}_k^T & \Delta\mathbf{u}_{k+1}^T & \dots & \Delta\mathbf{u}_{k+N_p-1}^T \end{bmatrix}^T \\
\mathbf{S} &= \begin{bmatrix} \mathbf{r}_{k+1} & \mathbf{r}_{k+2} & \dots & \mathbf{r}_{k+N_p} \end{bmatrix}^T \\
\mathbf{Y} &= \begin{bmatrix} \mathbf{y}_{k+1} & \mathbf{y}_{k+2} & \dots & \mathbf{y}_{k+N_p} \end{bmatrix}^T
\end{aligned}$$

Substituting (3.6) to (3.5), the state predictions for the system take the following

form:

$$\begin{aligned}
\mathbf{x}_{k+1} =& \tilde{\mathbf{A}}\mathbf{x}_k + \mathbf{D}\mathbf{r}_{k+1} + \mathbf{D}\boldsymbol{\beta}_1 \\
\mathbf{x}_{k+2} =& \tilde{\mathbf{A}}\mathbf{x}_{k+1} + \mathbf{D}\mathbf{r}_{k+2} + \mathbf{D}\boldsymbol{\beta}_2 \\
=& \tilde{\mathbf{A}}^2\mathbf{x}_k + \left(\tilde{\mathbf{A}}\mathbf{D}\mathbf{r}_{k+1} + \mathbf{D}\mathbf{r}_{k+2}\right) + \left(\tilde{\mathbf{A}}\mathbf{D}\boldsymbol{\beta}_1 + \mathbf{D}\boldsymbol{\beta}_2\right) \\
& \vdots \\
\mathbf{x}_{k+N_p} =& \tilde{\mathbf{A}}^{N_p}\mathbf{x}_k + \sum_{i=0}^{N_p-1} \tilde{\mathbf{A}}^i\mathbf{D}\mathbf{r}_{k+N_p-i} + \sum_{i=0}^{N_p-1} \tilde{\mathbf{A}}^i\mathbf{D}\boldsymbol{\beta}_{k+N_p-i}
\end{aligned}$$

or in a more compact matrix form:

$$\mathbf{X} = \mathbf{F}\mathbf{x}_k + \mathbf{H}_r\mathbf{S} + \mathbf{H}_r\boldsymbol{\beta}$$

in which

$$\mathbf{F} = \begin{bmatrix} \tilde{\mathbf{A}} \\ \tilde{\mathbf{A}}^2 \\ \vdots \\ \tilde{\mathbf{A}}^{N_p} \end{bmatrix} ; \mathbf{H}_r = \begin{bmatrix} \mathbf{D} & \mathbf{0}_1 & \cdots & \mathbf{0}_1 \\ \tilde{\mathbf{A}}\mathbf{D} & \mathbf{D} & \cdots & \mathbf{0}_1 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}}^{N_p-1}\mathbf{D} & \tilde{\mathbf{A}}^{N_p-2}\mathbf{D} & \cdots & \mathbf{D} \end{bmatrix} \tag{3.8}$$

where $\mathbf{0}_1$ is a $(n+1)p+(m-1)q \times p$ matrix of zeros. Next, the following matrices can be defined in order for the future control input vector to be written in a similar matrix form,

$$\tilde{\mathbf{K}}_1 = \begin{bmatrix} -\mathbf{K} \\ \mathbf{0}_2 \\ \mathbf{0}_2 \\ \vdots \\ \mathbf{0}_2 \end{bmatrix} ; \tilde{\mathbf{K}}_2 = \begin{bmatrix} \mathbf{0}_2 & \mathbf{0}_2 & \cdots & \mathbf{0}_2 & \mathbf{0}_2 \\ -\mathbf{K} & \mathbf{0}_2 & \cdots & \mathbf{0}_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & -\mathbf{K} & \cdots & \mathbf{0}_2 & \mathbf{0}_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_2 & \mathbf{0}_2 & \cdots & -\mathbf{K} & \mathbf{0}_2 \end{bmatrix} \tag{3.9}$$

where $\mathbf{K}$ is the control gain matrix defined in (3.4) and $\mathbf{0}_2$ is a zero matrix of

dimension $(n+1)p+(m-1)q \times q$. The future control input vector can now be written as:

$$\begin{aligned} \mathbf{U} &= \tilde{\mathbf{K}}_1 \mathbf{x}_k + \tilde{\mathbf{K}}_2 \mathbf{X} \\ &= \left(\tilde{\mathbf{K}}_1 + \tilde{\mathbf{K}}_2 \mathbf{F}\right) \mathbf{x}_k + \tilde{\mathbf{K}}_2 \mathbf{H}_r \mathbf{S} + \tilde{\mathbf{K}}_2 \mathbf{H}_r \boldsymbol{\beta} \end{aligned}$$

In the same manner, and noting that $\boldsymbol{\Delta}\mathbf{U} = -\mathbf{C}_1 \mathbf{u}_{k-1} + \mathbf{C}_2 \mathbf{U}$, with:

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0}_q \\ \mathbf{0}_q \\ \vdots \\ \mathbf{0}_q \\ \mathbf{0}_q \end{bmatrix} ; \mathbf{C}_2 = \begin{bmatrix} \mathbf{I}_q & \mathbf{0}_q & \mathbf{0}_q & \cdots & \mathbf{0}_q & \mathbf{0}_q \\ -\mathbf{I}_q & \mathbf{I}_q & \mathbf{0}_q & \cdots & \mathbf{0}_q & \mathbf{0}_q \\ \mathbf{0}_q & -\mathbf{I}_q & \mathbf{I}_q & \ddots & \mathbf{0}_q & \mathbf{0}_q \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0}_q & \mathbf{0}_q & \ddots & -\mathbf{I}_q & \mathbf{I}_q & \mathbf{0}_q \\ \mathbf{0}_q & \mathbf{0}_q & \cdots & \mathbf{0}_q & -\mathbf{I}_q & \mathbf{I}_q \end{bmatrix} \tag{3.10}$$

the future control increment vector is written as:

$$\boldsymbol{\Delta}\mathbf{U} = \mathbf{C}_1 u_{k-1} + \mathbf{C}_2 \left(\tilde{\mathbf{K}}_1 + \tilde{\mathbf{K}}_2 \mathbf{F}\right) \mathbf{x}_k + \mathbf{C}_2 \tilde{\mathbf{K}}_2 \mathbf{H}_r \mathbf{S} + \mathbf{C}_2 \tilde{\mathbf{K}}_2 \mathbf{H}_r \boldsymbol{\beta}$$

Finally, defining the $pN_p \times ((n+1)p+(m-1)q)N_p$ block diagonal matrix $\bar{\mathbf{C}}$, with $\mathbf{C}$ on it's diagonal, the predicted output vector is written:

$$\begin{aligned} \mathbf{Y} &= \bar{\mathbf{C}} \mathbf{X} \\ &= \bar{\mathbf{C}} \mathbf{F} \mathbf{x}_k + \bar{\mathbf{C}} \mathbf{H}_r \mathbf{S} + \bar{\mathbf{C}} \mathbf{H}_r \boldsymbol{\beta} \end{aligned}$$

At this point, it is useful to note that the double inequalities of (3.7b) can be written as two single inequalities and in matrix form as for example for the first

one:

$$-\mathbf{U} \leq -\underline{\mathbf{U}} \tag{3.11a}$$

$$\mathbf{U} \leq \overline{\mathbf{U}} \tag{3.11b}$$

where $\underline{\mathbf{U}}$ and $\overline{\mathbf{U}}$ are vectors with the minimum and maximum allowed values for $\mathbf{u}$. The second and third double inequalities of (3.7b) can also be described in matrix form as in (3.11) with $\underline{\Delta\mathbf{U}}$, $\overline{\Delta\mathbf{U}}$, $\underline{\mathbf{Y}}$ and $\overline{\mathbf{Y}}$ appropriately defined. Now, the control problem (3.7) can take the following matrix form:

$$\min_{\boldsymbol{\beta}} \quad \boldsymbol{\beta}^T\boldsymbol{\beta} \tag{3.12a}$$

$$\text{subject to:} \quad \mathbf{M}\boldsymbol{\beta} \leq \mathbf{N} \tag{3.12b}$$

where,

$$\mathbf{M} = \begin{bmatrix} -\tilde{\mathbf{K}}_2\mathbf{H}_r \\ \tilde{\mathbf{K}}_2\mathbf{H}_r \\ -\mathbf{C}_2\tilde{\mathbf{K}}_2\mathbf{H}_r \\ \mathbf{C}_2\tilde{\mathbf{K}}_2\mathbf{H}_r \\ -\bar{\mathbf{C}}r_k\mathbf{H}_r \\ \bar{\mathbf{C}}r_k\mathbf{H}_r \end{bmatrix} ; \mathbf{N} = \begin{bmatrix} -\underline{\mathbf{U}} + \left(\tilde{\mathbf{K}}_1 + \tilde{\mathbf{K}}_2\mathbf{F}\right)\mathbf{x}_k + \tilde{\mathbf{K}}_2\mathbf{H}_r\mathbf{S} \\ \overline{\mathbf{U}} - \left(\tilde{\mathbf{K}}_1 + \tilde{\mathbf{K}}_2\mathbf{F}\right)\mathbf{x}_k - \tilde{\mathbf{K}}_2\mathbf{H}_r\mathbf{S} \\ -\underline{\Delta\mathbf{U}} - \mathbf{C}_1u_{k-1} + \mathbf{C}_2\left(\tilde{\mathbf{K}}_1 + \tilde{\mathbf{K}}_2\mathbf{F}\right)\mathbf{x}_k + \mathbf{C}_2\tilde{\mathbf{K}}_2\mathbf{H}_r\mathbf{S} \\ \overline{\Delta\mathbf{U}} + \mathbf{C}_1u_{k-1} - \mathbf{C}_2\left(\tilde{\mathbf{K}}_1 + \tilde{\mathbf{K}}_2\mathbf{F}\right)\mathbf{x}_k - \mathbf{C}_2\tilde{\mathbf{K}}_2\mathbf{H}_r\mathbf{S} \\ -\underline{\mathbf{Y}} + \bar{\mathbf{C}}\mathbf{F}\mathbf{x}_k + \bar{\mathbf{C}}\mathbf{H}_r\mathbf{S} \\ \overline{\mathbf{Y}} - \bar{\mathbf{C}}\mathbf{F}\mathbf{x}_k - \bar{\mathbf{C}}\mathbf{H}_r\mathbf{S} \end{bmatrix}$$

and the inequality sign refers to element by element inequalities.

This is now clearly in the form of (2.4) with $\mathbf{H} = \mathbf{I}_{N_p}$, $\mathbf{p} = \mathbf{0}$ and no equality constraints and can therefore be solved using any of the existing constraint optimisation techniques. In this regard, the quadprog function of the MATLAB® optimisation toolbox is evaluated that uses an active set method (see Section 2.2.3) or an interior point (see Section 2.2.4) to solve the optimisation program (depending on the problem size). Finally, the RHC approach is used and the problem is

solved at every sampling instant.

3.3 Stability and Reference Tracking

Since a stabilising control law is already applied to the system, stability of the control scheme is trivial. In the presence of constraints, the RG produces a sequence of reference signal perturbations that leads to constraint satisfaction and therefore stability is guaranteed whilst satisfying the constraints.

A common approach to ensure type one servomechanism performance for the RG controlled system is to force the β parameters to zero after a specific point within the prediction horizon (i.e. the reference that would be applied is the actual one and the PIP controller would ensure type one performance). Another approach to ensure reference tracking is to parametrise the perturbation as $\tilde{\beta}_k = \beta_k \mu_k$, where μ_k is monotonically decreasing with k and β_k is bounded, which is in principle the same as forcing β to zero (yet in a smoother way). This is based on the approach by Bemporad et al. (1997) where a different (but decaying) parametrisation of the perturbation is used. However, in this thesis it is assumed that reference tracking is practically achieved though it is clear that the approach can be adequately changed to force a decaying β.

3.4 Simulation example

To present the characteristics of the RG technique presented above, the following reference tracking control problem is considered. Let a marginally stable, non–minimum phase Single Input Single Output (SISO) system be defined by the following difference equation:

$$y_k - 1.7y_{k-1} + y_{k-2} = -0.5u_{k-1} + 2u_{k-2}$$

The control objective is to track constant reference signals while the input signal has a maximum magnitude of 2 units (i.e. $|u| \leq 2$) and the rate of change of the input signal is of magnitude less than 0.5 units/sample (i.e. $|\Delta u| \leq 0.5$). Both the input and it's rate–of–change are considered to saturate at their boundary values. Initially, the system is expressed by the NMSS state space form described in Section 3.1, with the state vector being:

$$\mathbf{x}_k = \begin{bmatrix} y_k & y_{k-1} & u_{k-1} & z_k \end{bmatrix}^T$$

Then, a PIP controller is designed using pole assignment and to highlight the difference between the RG technique and regular PIP control, the closed loop is designed to have a fast response to reference changes. Namely, two of the poles are placed in the origin while the other two at 0.2 of the z-plane. This results in a state feedback controller of the form of (3.4) with the following gain vector.

$$\mathbf{k} = \begin{bmatrix} 3.16 & -3.05 & 6.10 & -0.64 \end{bmatrix}$$

At this point, it should be noted that, to avoid integral windup, the implementation of the PIP controller is based on it's incremental form (see Taylor et al., 2004a), according to which, the control signal is given by:

$$\mathbf{u}_k = \mathbf{u}_{k-1} - \mathbf{K}\Delta\mathbf{x}_k \tag{3.13a}$$

with the following corrections:

$$\mathbf{u}_k = \begin{cases} \mathbf{u}_{k-1} + \overline{\Delta \mathbf{u}} & , \; -\mathbf{K}\mathbf{x}_k > \overline{\Delta \mathbf{u}} \\ \mathbf{u}_{k-1} - \underline{\Delta \mathbf{u}} & , \; -\mathbf{K}\mathbf{x}_k < \underline{\Delta \mathbf{u}} \end{cases} \quad \text{and} \quad \mathbf{u}_k = \begin{cases} \overline{\mathbf{u}} & , \; \mathbf{u}_k > \overline{\mathbf{u}} \\ \underline{\mathbf{u}} & , \; \mathbf{u}_k < \underline{\mathbf{u}} \end{cases} \tag{3.13b}$$

A simulation of the closed loop system for two reference changes (one *small* from 0 units to 5 units at the 5th sampling instant and one *large* from 5 units

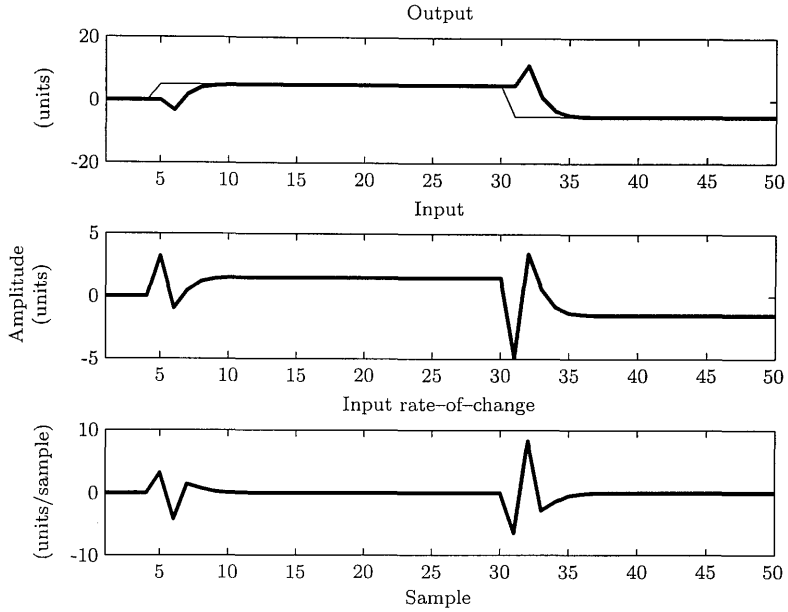

Figure 3.2: System output (top figure), input (middle figure) and input rate–of–change (bottom figure) for the case where the constraints are not present. In the top figure the reference signal (thin line) is also shown.

to -5 units at the 31st sampling instant) when no constraints are applied to the system is shown in Figure 3.2 (the initial condition for the system is considered to be $\mathbf{x}_0 = \mathbf{0}$). It is clear that both the input signal and it's rate–of–change violate the constraints defined by the system specifications. In order to systematically deal with the constraints, a RG is designed for the closed loop system.

Using a prediction horizon of $N_p = 10$ for the RG, the system is again simulated for the same reference signals as before, with saturating constraints on the control input signal and it's rate–of–change, and the same initial conditions. The result is depicted in Figure 3.3 for both the system with and without the RG. It is clear that the system with the RG manages to overcome the unwanted oscillations that appear in the *small* reference change and results in a stable response in the case of the *large* reference change (in the latter case, the system without the RG produces an unstable response).

Although the response of both systems seems to be very close until they initially

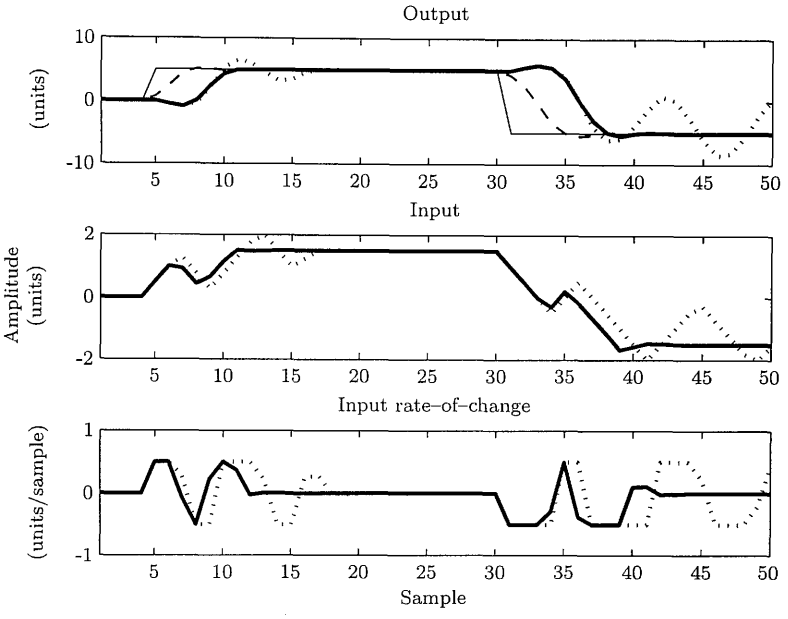

Figure 3.3: Output (top figure), input (middle figure) and input rate–of–change (bottom figure) for the system with (thick solid lines) and without (dotted lines) the RG. In the top figure the actual (thin solid line) and the artificial (thin dashed line) reference signals are also shown.

reach the set point, the system without the reference governor fails to remain close to it. It fluctuates before settling (*small* reference change) or even goes unstable in the case of the *large* reference change. This is directly related to closed loop system dynamics and the saturation of the control signal and it's rate–of–change. In this example an initially very fast closed loop system was designed to emphasise this phenomenon. Both systems hit the constraints when the set point change occurs, but in the case of the system with the RG the reference signal is smoothened in order for the controller to approach the actual set–point slightly slower and avoid unnecessary oscillations or instability. It is therefore evident that the introduction of the RG can improve the system response of a closed loop system in the presence of saturating constraints.

To show the effect of the prediction horizon N_p, the case of a much shorter and a much longer prediction horizon (namely $N_p = 3$ and $N_p = 30$) is presented in Figure 3.4. As can be seen, a longer prediction horizon results in smother handling

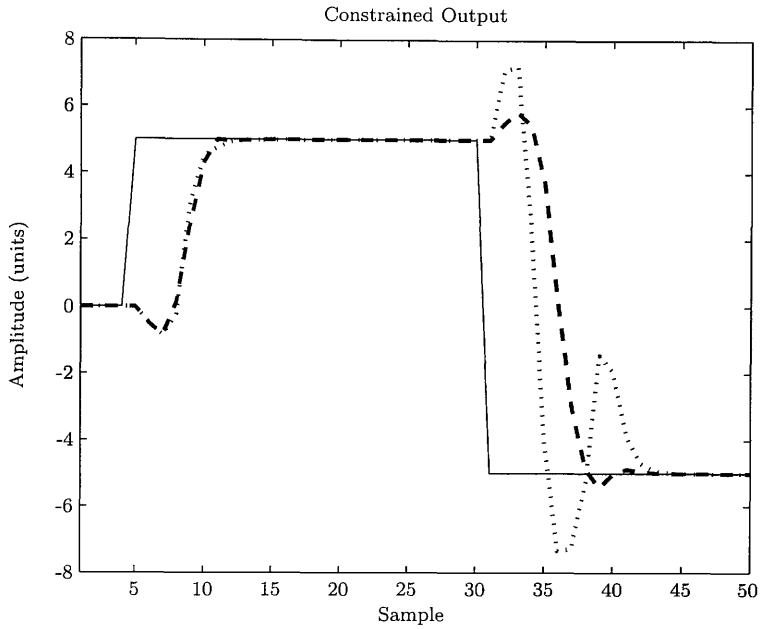

Figure 3.4: Comparison of the system response for the cases where $N_p = 3$ (dotted line) and $N_p = 30$ (dashed line). The actual reference signal is shown as a solid line.

of the nonlinearity caused by the constraint and tends to decrease the overshoot in the *large* reference change. In the *small* reference change the result is not noticeable since the available dof are enough for a smooth response. At this point, it should be stressed that even in the case of the very small prediction horizon of $N_p = 3$ the RG system produces a stable system output. It should also be noted that the response for the cases of $N_p = 10$ and $N_p = 30$ is indistinguishable. This is because a prediction horizon of 10 is already long enough for the RG to foresee and avoid any constraint violation (as is for the small reference change between the RG with $N_p = 3$ and $N_p = 30$ shown in Figure 3.3). However, a larger set–point change might require a longer prediction horizon and in a practical situation it should be chosen based on experience and knowledge on the expected set–point changes.

The resulting β parameter for the short and long prediction horizons is shown in Figure 3.5 and as already mentioned it can provide useful insight on the effect

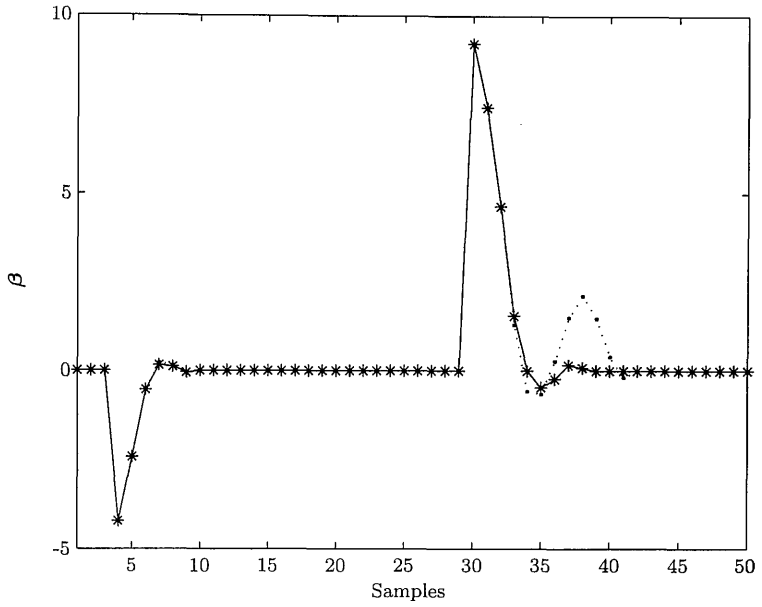

Figure 3.5: The reference perturbations for the RG controlled system with $N_p = 3$ (dotted line) and $N_p = 30$ (solid-star line).

of the disturbances to the controlled system. By the values of β and (3.6) it is can be seen that the slack reference signal is brought towards the actual system output and subsequently moved closer to the next set–point. This is the case for both set–point changes. However, in the *large* set–point change, the RG with the longer prediction horizon has already foreseen the need for the controller to slow down and therefore the oscillations of β around zero are minimised.

3.5 Concluding remarks

This chapter has presented the RG technique for constraint handling when a stabilising controller is already present. This controller is used to pre–stabilise (if necessary) the open loop system and fulfil the design requirements in the constraint free case. The RG is then applied to change the reference signal in order for the closed loop system to satisfy the constraints.

Although the RG approach to constraint handling is not new, the parametri-

sation of the problem that has been developed in this thesis resembles the one used in the CLP approach rather than other RG approaches. This is mainly due to it's simplicity in implementation and the insight it can provide on the effect of the constraints to the controlled system. Concerning reference tracking, it is common for RG approaches to require for the perturbation around the reference signal to decay (and eventually reach zero), but in this case it is argued that once the set–point is within the constrained region, it will be reached and the resulting perturbation will be minimal (since the square of the perturbations is minimised). Nevertheless, a methodology is sketched in order to formally provide reference tracking.

The simple principle that underlies the RG technique makes it an attractive tool for non–linear systems (a detailed study of the RG for non–linear systems is presented by Bemporad, 1998a). In this regard, this chapter also serves as an introduction to the notion of the RG that will later be applied to the non–linear class of SDP models (Chapter 8).

Chapter 4

MPC with an integral–of–error state

The dependence of Model Predictive Controllers on the estimated model of the system is of particular importance (e.g. Rossiter, 2003). In this regard, as already mentioned, Wang and Young (2006) have recently proposed an approach to MPC based on the definition of a NMSS model. This approach eliminates the need for an observer such as a deterministic state reconstructor or a stochastic Kalman filter.

Such NMSS models have previously been used for linear (fixed gain) PIP control system design (e.g. Young et al., 1987; Taylor et al., 2000b). As already presented in Chapter 3, for NMSS/PIP control, the state vector consists only of the present and past values of the output variable, past values of the input variable and an integral–of–error state variable, introduced to ensure type one servomechanism performance. All these state variables are directly measurable, making the controller potentially less sensitive to the problems of model mismatch than the minimal state equivalent using an observer.

Wang and Young (2006) develop an approach for inherent handling of system and actuator constraints for NMSS models using MPC methods. Furthermore, they show that the NMSS/MPC approach leads to improved performance (re-

garding the chosen performance objectives) in comparison to conventional *minimal* MPC with an observer. However, in order to ensure steady state tracking, Wang and Young (2006) defined a state vector consisting of the *differences* of past output and input values, an approach that is very common in chemical and process control industries. Nevertheless, since a number of useful tuning techniques have been developed for the NMSS/PIP form, including multi–objective optimisation of the linear quadratic cost function weights (Tych, 1994; Tych et al., 1996; Chotai et al., 1998), the present chapter delivers an MPC control scheme using this alternative framework with an explicit integral–of–error state variable.

This new approach is formed to serve as a starting point for experience gained in PIP design to be adopted within the NMSS/MPC framework. In this direction, it is later shown in Chapter 5 that this particular formulation of the NMSS/MPC controller provides the designer with extra freedom (that is a result of the form of the state vector and the integral–of–error state variable) to deal with performance requirements.

In the following, Section 4.1 presents the NMSS/MPC controller of Wang and Young (2006) for comparison and future reference while Section 4.2 presents the alternative controller formulation proposed by the author and Section 4.3 develops it's disturbance rejection and set–point following properties along with stability results. Then Section 4.4 presents it's properties using simulation examples while Section 4.5 summarises the conclusions of this chapter.

4.1 The controller of Wang and Young (2006)

This section presents briefly the NMSS system description and controller formulation that has initially been described by Wang and Young (2006). Then, Section 4.2 presents an alternative NMSS/MPC controller based on the state space description of (3.1), used for PIP control and the state vector (3.2).

4.1.1 System Description

Consider the difference equation description (2.1) of a q–input p–output system. It can also be represented in terms of *differenced* input and output variables yielding embedded integral action in the MPC system as:

$$\mathbf{\Delta y}_k+\mathbf{A}_1\mathbf{\Delta y}_{k-1}+\ldots+\mathbf{A}_n\mathbf{\Delta y}_{k-n} = \mathbf{B}_1\mathbf{\Delta u}_{k-1}+\mathbf{B}_2\mathbf{\Delta u}_{k-2}+\ldots+\mathbf{B}_m\mathbf{\Delta u}_{k-m} \quad (4.1)$$

where $\mathbf{\Delta}$ is the difference operator (i.e. $\mathbf{\Delta y}_k = \mathbf{y}_k - \mathbf{y}_{k-1}$). Defining the NMSS state vector as:

$$\mathbf{x}_{\Delta,k}^T = \begin{bmatrix} \mathbf{\Delta y}_k^T & \mathbf{\Delta y}_{k-1}^T & \cdots & \mathbf{\Delta y}_{k-n+1}^T & \mathbf{\Delta u}_{k-1}^T & \mathbf{\Delta u}_{k-2}^T & \cdots & \mathbf{\Delta u}_{k+m-1}^T \end{bmatrix} \quad (4.2)$$

where the subscript $_\Delta$ refers to the fact that the state vector is defined using the differenced values of inputs and outputs, while the subscript $_k$ refers to the sampling instant. The state space description of the system has the following form:

$$\mathbf{\Delta x}_{\Delta,k} = \mathbf{A}_\Delta\mathbf{\Delta x}_{\Delta,k-1} + \mathbf{B}_\Delta\Delta u_{k-1}$$

$$\mathbf{\Delta y}_k = \mathbf{C}_\Delta\mathbf{\Delta x}_{\Delta,k}$$

The matrices in the above state space description are:

$$\mathbf{A}_\Delta = \begin{bmatrix} -\mathbf{A}_1 & \cdots & -\mathbf{A}_{n-1} & -\mathbf{A}_n & \mathbf{B}_2 & \cdots & \mathbf{B}_{m-1} & \mathbf{B}_m \\ \mathbf{I}_p & \cdots & \mathbf{0}_p & \mathbf{0}_p & \mathbf{0}_{pq} & \cdots & \mathbf{0}_{pq} & \mathbf{0}_{pq} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_p & \cdots & \mathbf{I}_p & \mathbf{0}_p & \mathbf{0}_{pq} & \cdots & \mathbf{0}_{pq} & \mathbf{0}_{pq} \\ \mathbf{0}_{qp} & \cdots & \mathbf{0}_{qp} & \mathbf{0}_{qp} & \mathbf{0}_q & \cdots & \mathbf{0}_q & \mathbf{0}_q \\ \mathbf{0}_{qp} & \cdots & \mathbf{0}_{qp} & \mathbf{0}_{qp} & \mathbf{I}_q & \cdots & \mathbf{0}_q & \mathbf{0}_q \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{qp} & \cdots & \mathbf{0}_{qp} & \mathbf{0}_{qp} & \mathbf{0}_q & \cdots & \mathbf{I}_q & \mathbf{0}_q \end{bmatrix}; \mathbf{B}_\Delta = \begin{bmatrix} \mathbf{B}_1^T \\ \mathbf{0}_{qp} \\ \vdots \\ \mathbf{0}_{qp} \\ \mathbf{I}_q \\ \mathbf{0}_q \\ \vdots \\ \mathbf{0}_q \end{bmatrix} \quad (4.3a)$$

and

$$\mathbf{C}_\Delta = \begin{bmatrix} \mathbf{I}_p & \mathbf{0}_p & \cdots & \mathbf{0}_p & \mathbf{0}_p & \mathbf{0}_{pq} & \cdots & \mathbf{0}_{pq} \end{bmatrix} \tag{4.3b}$$

Subsequently, the augmented state vector is defined as:

$$\mathbf{x}_k^T = \begin{bmatrix} \Delta\mathbf{x}_{\Delta,k}^T & \mathbf{y}_k^T \end{bmatrix} \tag{4.4}$$

The NMSS representation of the system that is used for state and output predictions in the MPC algorithm is then defined as:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{A}_\Delta & \mathbf{0} \\ \mathbf{C}_\Delta\mathbf{A}_\Delta & \mathbf{I}_p \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} \mathbf{B}_\Delta \\ \mathbf{C}_m\mathbf{B}_\Delta \end{bmatrix} \Delta\mathbf{u}_{k-1} \tag{4.5a}$$

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{0}^T & \mathbf{I}_p \end{bmatrix} \mathbf{x}_k \tag{4.5b}$$

where $\mathbf{0}$ is a $np + (m-1)q \times p$ matrix of zeros. For notational simplicity, in the sequel the matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are defined as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_\Delta & \mathbf{0} \\ \mathbf{C}_\Delta\mathbf{A}_\Delta & \mathbf{I}_p \end{bmatrix}; \qquad \mathbf{B} = \begin{bmatrix} \mathbf{B}_\Delta \\ \mathbf{C}_m\mathbf{B}_\Delta \end{bmatrix} \qquad \text{and } \mathbf{C} = \begin{bmatrix} \mathbf{0}^T & \mathbf{I}_p \end{bmatrix} \tag{4.6}$$

4.1.2 Control Description

Following the standard MPC approach, the future state, future output and control increment vectors are defined as:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{k+1}^T & \mathbf{x}_{k+2}^T & \cdots & \mathbf{x}_{k+N_p}^T \end{bmatrix}^T \tag{4.7a}$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_{k+1}^T & \mathbf{y}_{k+2}^T & \cdots & \mathbf{y}_{k+N_p}^T \end{bmatrix}^T \tag{4.7b}$$

$$\Delta\mathbf{U} = \begin{bmatrix} \Delta\mathbf{u}_k^T & \Delta\mathbf{u}_{k+1}^T & \cdots & \Delta\mathbf{u}_{k+N_c-1}^T \end{bmatrix}^T \tag{4.7c}$$

where N_p and N_c are the prediction and control horizons respectively as described in Section 2.3. The future state, the future control increment, future output and the current state vectors, are related by the following equations:

$$\mathbf{X} = \mathbf{F}\mathbf{x}_k + \mathbf{\Phi}\mathbf{\Delta U}$$

$$\mathbf{Y} = \bar{\mathbf{C}}\mathbf{X}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^{N_c} \\ \mathbf{A}^{N_c+1} \\ \vdots \\ \mathbf{A}^{N_p} \end{bmatrix} ; \mathbf{\Phi} = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{N_c-1}\mathbf{B} & \mathbf{A}^{N_c-2}\mathbf{B} & \cdots & \mathbf{B} \\ \mathbf{A}^{N_c}\mathbf{B} & \mathbf{A}^{N_c-1}\mathbf{B} & \cdots & \mathbf{AB} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}^{N_p-1}\mathbf{B} & \mathbf{A}^{N_p-2}\mathbf{B} & \cdots & \mathbf{A}^{N_p-N_c}\mathbf{B} \end{bmatrix} \tag{4.8}$$

and $\bar{\mathbf{C}}$ is a N_p block diagonal matrix of dimensions $pN_p \times ((n+1)p+(m-1)q)N_p$, with the matrix $\mathbf{C}$ on it's diagonal. The future reference trajectory vector is subsequently defined as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{r}_{k+1}^T & \mathbf{r}_{k+2}^T & \cdots & \mathbf{r}_{k+N_p}^T \end{bmatrix}^T \tag{4.9}$$

The control law is then defined by minimising the following cost:

$$J = (\mathbf{S} - \mathbf{Y})^T \bar{\mathbf{Q}} (\mathbf{S} - \mathbf{Y}) + \mathbf{\Delta U}^T \bar{\mathbf{R}} \mathbf{\Delta U} \tag{4.10}$$

at each sampling instant, subject to constraints on the inputs, outputs and the rate–of–change of the input signals. In the above cost function, the $\bar{\mathbf{Q}}$ is a $pN_p \times pN_p$ positive definite matrix, and $\bar{\mathbf{R}}$ is a $qN_c \times qN_c$ positive semi–definite matrix.

More information on the above NMSS/MPC description, a detailed discription of it's properties and a methodology to formulate the constraints in the required matrix form can be found in Wang and Young (2006).

4.2 The alternative NMSS/MPC controller

This section presents an alternative MPC formulation that is based on the NMSS system description that is used for PIP control. Specifically, the system described by the state space description (3.1) defined in Section 3.1.

In a similar manner to PIP Linear Quadratic optimal control, the following index is defined:

$$J(\mathbf{x}_k, \mathbf{r}_k) = \sum_{i=1}^{N_p} \mathbf{x}_{k+i}^T \mathbf{Q} \mathbf{x}_{k+i} + \sum_{i=0}^{N_c-1} \mathbf{u}_{k+i}^T \mathbf{R} \mathbf{u}_{k+i} \tag{4.11}$$

where $\mathbf{x}_{k+i}$, $i = 1, \ldots, N_p$ are the predicted state vectors (based on the system evolution of (3.1)) and $\mathbf{u}_{k+i}$, $i = 0, \ldots, N_c - 1$ are the predicted control vectors, which are also the variables to be optimised.

Remark 4.2.1. *Although the dependence of J on $\mathbf{r}_k$ is not explicitly shown in* (4.11)*, it is implicit since the state depends on $\mathbf{r}_k$ and it's evolution through the integral–of–error state. This dependence is merely stated to point out that the cost function is related to the set point and as will be shown in Section 4.3 the proposed control scheme results in a closed loop*[1] *system that follows any constant reference without any error.*

The control action is therefore derived by numerically solving the following optimisation problem:

$$\min_{\mathbf{u}_k, \ldots, \mathbf{u}_{k+N_c-1}} \quad J(\mathbf{x}_k, \mathbf{r}_k) \tag{4.12a}$$

[1]The term closed loop is used at this point to refer to the controlled system. As is also depicted in Figure 4.1, output information is fed back to the controller in order for the next control signal to be calculated, leading to a closed loop formulation.

$$\text{subject to:} \begin{cases} \underline{\mathbf{u}} \leq \mathbf{u}_{k+i} \leq \overline{\mathbf{u}} & , i = 0, \ldots, N_c - 1 \\ \underline{\Delta \mathbf{u}} \leq \Delta \mathbf{u}_{k+i} \leq \overline{\Delta \mathbf{u}} & , i = 0, \ldots, N_c - 1 \\ \underline{\mathbf{y}} \leq \mathbf{y}_{k+i} \leq \overline{\mathbf{y}} & , i = 1, \ldots, N_p \end{cases} \tag{4.12b}$$

where $\Delta \mathbf{u}_{k+i} = \mathbf{u}_{k+i} - \mathbf{u}_{k+i-1}$ is the vector of control increments at each sampling instant; $\underline{\cdot}$ and $\overline{\cdot}$ refer to minimum and maximum allowed values for the system variables; and the inequalities in the constraints are element by element inequalities. As in Section 3.2, for commercial optimisation tools to be used, the optimisation problem needs to be cast into a more compact matrix form. In this regard, the predicted state, future output, future control input increment and future reference are defined as in (4.7) and (4.9) respectively and the future control action as:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_k^T & \mathbf{u}_{k+1}^T & \ldots & \mathbf{u}_{k+N_c-1}^T \end{bmatrix}^T \tag{4.13}$$

where it should be noted that the vector $\mathbf{U}$ is also the vector of the optimisation parameters.

From this point on, it is assumed that the control signal retains it's last value after the control horizon (i.e. $\mathbf{u}_{k+i} = \mathbf{u}_{k+N_c-1}$, $i = N_c, \ldots, N_p$), although other alternatives are clearly possible (e.g. it can be forced to obtain it's steady state value for a given set–point at the end of the control horizon and retain that value).

By recursive application of (3.1a), the predicted system evolution is:

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{D}\mathbf{r}_{k+1} \\
\mathbf{x}_{k+2} &= \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{u}_{k+1} + \mathbf{D}\mathbf{r}_{k+2} \\
&= \mathbf{A}^2\mathbf{x}_k + (\mathbf{A}\mathbf{B}\mathbf{u}_k + \mathbf{B}\mathbf{u}_{k+1}) + (\mathbf{A}\mathbf{D}\mathbf{r}_{k+1} + \mathbf{D}\mathbf{r}_{k+2}) \\
&\vdots \\
\mathbf{x}_{k+N_c} &= \mathbf{A}^{N_c}\mathbf{x}_k + \sum_{i=0}^{N_c-1} \mathbf{A}^i\mathbf{B}\mathbf{u}_{k+N_c-1-i} + \sum_{i=0}^{N_c-1} \mathbf{A}^i\mathbf{D}\mathbf{r}_{k+N_c-i} \\
&\vdots \\
\mathbf{x}_{k+N_p} &= \mathbf{A}^{N_p}\mathbf{x}_k + \sum_{i=N_p-N_c}^{N_p-1} \mathbf{A}^i\mathbf{B}\mathbf{u}_{k+N_p-1-i} + \left(\sum_{i=0}^{N_p-N_c-1} \mathbf{A}^i\mathbf{B} \right) \mathbf{u}_{k+N_c-1} + \\
&\quad + \sum_{i=0}^{N_p-1} \mathbf{A}^i\mathbf{D}\mathbf{r}_{k+N_p-i}
\end{aligned}
$$

and the prediction equations can take the following matrix form:

$$\mathbf{X} = \mathbf{F}\mathbf{x}_k + (\mathbf{\Phi} + \mathbf{\Phi}_1)\,\mathbf{U} + \mathbf{H}_r\mathbf{S} \tag{4.14}$$

where the matrices $\mathbf{F}$ and $\mathbf{H}_r$ that represent the effect of the current state and reference signal to the predicted states are defined as in (3.8) and $\mathbf{\Phi}$ is given by (4.8). It should be noted here that the matrix which represents the effect of the control inputs to the predicted states is split into $\mathbf{\Phi}$ and $\mathbf{\Phi}_1$ to distinguish between the effect of the control inputs until the end the control horizon and the one due to the application of the last control input into the rest of the prediction horizon.

The $\mathbf{\Phi}_1$ matrix is defined as follows:

$$\mathbf{\Phi}_1 = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}+\mathbf{AB} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}+\mathbf{AB}+\mathbf{A}^2\mathbf{B} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}+\cdots+\mathbf{A}^{N_p-N_c-1}\mathbf{B} \end{bmatrix}$$

The function to be minimised can subsequently take the following matrix form:

$$J_M = \mathbf{X}^T\bar{\mathbf{Q}}\mathbf{X} + \mathbf{U}^T\bar{\mathbf{R}}\mathbf{U} \tag{4.15a}$$

$$= \frac{1}{2}\mathbf{U}^T\left(\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\tilde{\mathbf{\Phi}} + \bar{\mathbf{R}}\right)\mathbf{U} + \left(\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\left(\mathbf{F}\mathbf{x}_k + \mathbf{H}_r\mathbf{S}\right)\right)^T\mathbf{U} + const. \tag{4.15b}$$

where $\tilde{\mathbf{\Phi}} = \mathbf{\Phi} + \mathbf{\Phi}_1$; $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ are block diagonal matrices with $\mathbf{Q}$ and $\mathbf{R}$ on their diagonals respectively; Equation (4.15b) is derived by substitution of (4.14) to (4.15a) and *const.* in (4.15b) refers to elements that are not affected by the choice of $\mathbf{U}$ and therefore are not of interest in the optimisation procedure.

Finally, in order to represent the output and control increment constraints in matrix form, the output prediction ($\mathbf{Y} = \bar{\mathbf{C}}\mathbf{X}$) and control increment prediction equations can be written as:

$$\mathbf{Y} = \bar{\mathbf{C}}\mathbf{F}\mathbf{x}_k + \bar{\mathbf{C}}\tilde{\mathbf{\Phi}}\mathbf{U} + \bar{\mathbf{C}}\mathbf{H}_r\mathbf{S}$$

$$\mathbf{\Delta U} = -\mathbf{C}_1\mathbf{u}_{k-1} + \mathbf{C}_2\mathbf{U}$$

where the matrix $\bar{\mathbf{C}}$ is a block diagonal matrix with $\mathbf{C}$ on it's diagonal and the matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ are defined as in (3.10). Therefore, and following a procedure similar to the one presented in Section 3.2 for the constraint inequalities (i.e. con-

verting each double inequality to two single ones as in (3.11)), the optimisation problem takes the following form:

$$\min_{\mathbf{U}} \quad \frac{1}{2}\mathbf{U}^T\left(\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\tilde{\mathbf{\Phi}}+\bar{\mathbf{R}}\right)\mathbf{U}+\left(\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\left(\mathbf{F}\mathbf{x}_k+\mathbf{H}_r\mathbf{S}\right)\right)^T\mathbf{U} \tag{4.16a}$$

$$\text{subject to:} \quad \mathbf{M}\mathbf{U}\leq\mathbf{N} \tag{4.16b}$$

where,

$$\mathbf{M}=\begin{bmatrix}-\mathbf{I}_{N_c}\\ \mathbf{I}_{N_c}\\ -\mathbf{C}_2\\ \mathbf{C}_2\\ -\bar{\mathbf{C}}\tilde{\mathbf{\Phi}}\\ \bar{\mathbf{C}}\tilde{\mathbf{\Phi}}\end{bmatrix};\ \mathbf{N}=\begin{bmatrix}-\underline{\mathbf{U}}\\ \overline{\mathbf{U}}\\ -\underline{\mathbf{\Delta U}}-\mathbf{C}_1u_{k-1}\\ \overline{\mathbf{\Delta U}}+\mathbf{C}_1u_{k-1}\\ -\underline{\mathbf{Y}}+\bar{\mathbf{C}}\mathbf{F}\mathbf{x}_k+\bar{\mathbf{C}}\mathbf{H}_r\mathbf{S}\\ \overline{\mathbf{Y}}-\bar{\mathbf{C}}\mathbf{F}\mathbf{x}_k-\bar{\mathbf{C}}\mathbf{H}_r\mathbf{S}\end{bmatrix} \tag{4.16c}$$

and the inequality in (4.16b) refers to element by element inequalities.

It is clear that the above optimisation problem is in the form (2.4) with $\mathbf{H}=\left(\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\tilde{\mathbf{\Phi}}+\bar{\mathbf{R}}\right)$ and $\mathbf{p}=\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\left(\mathbf{F}\mathbf{x}_k+\mathbf{H}_r\mathbf{S}\right)$. The control vector $\mathbf{u}_k$ is subsequently derived by solving the problem (4.16) at every sampling instant and is applied to the system following the Receding Horizon technique presented in Section 2.3.

4.3 Properties of the control scheme

4.3.1 Set–point following and disturbance rejection

This section exploits the block diagram form of the alternative MPC formulation presented in the previous section and delivers the required steady state tracking of the reference input, along with it's disturbance rejection properties. It should be noted that the *unconstrained* case is considered, while Remark 4.3.1 extends the results to the general case where constraints are present.

The unconstrained solution to the problem (4.16) is derived by direct differentiation of (4.15b) (it should be noted that the second derivative of J_M over U is non–negative resulting in a minimum of J_M (see Gill et al., 1981, p. 61)). Omitting the details for brevity, the solution has the fixed gain controller form:

$$\mathbf{U} = -\mathbf{K_s}\mathbf{x}(k) - \mathbf{R_s}\mathbf{S} \tag{4.17}$$

where,

$$\mathbf{K}_s = \left(\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\tilde{\mathbf{\Phi}} + \bar{\mathbf{R}}\right)^{-1}\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\mathbf{F}$$

$$\mathbf{R}_s = \left(\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\tilde{\mathbf{\Phi}} + \bar{\mathbf{R}}\right)^{-1}\tilde{\mathbf{\Phi}}^T\bar{\mathbf{Q}}\mathbf{H}_r$$

Letting the first qN_c rows of $\mathbf{K}_s$ and $\mathbf{R}_s$ be:

$$\begin{aligned}\mathbf{K}_s^{(q)} &= \begin{bmatrix}\mathbf{K}_1 & \cdots & \mathbf{K}_n & \mathbf{K}_{n+1} & \cdots & \mathbf{K}_{n+m-1} & \mathbf{K}_{n+m}\end{bmatrix}\\ \mathbf{R}_s^{(q)} &= \begin{bmatrix}\mathbf{R}_1 & \cdots & \mathbf{R}_{N_p}\end{bmatrix}\end{aligned}$$

and exploiting the structure of the state vector $\mathbf{x}_k$, the control law is written:

$$\begin{aligned}\mathbf{u}_k = &-\left(\mathbf{K}_1\mathbf{y}_k + \mathbf{K}_2\mathbf{y}_{k-1} + \cdots + \mathbf{K}_n\mathbf{y}_{k-n+1}\right)\\ &-\left(\mathbf{K}_{n+1}\mathbf{u}_{k-1} + \cdots + \mathbf{K}_{n+m-1}\mathbf{u}_{k-m+1}\right)\\ &-\left(\mathbf{R}_1\mathbf{r}_{k+1} + \mathbf{R}_2\mathbf{r}_{k+2}\cdots + \mathbf{R}_{N_p}\mathbf{r}_{k+N_p}\right)\\ &-\mathbf{K}_{n+m}\mathbf{z}_k\end{aligned}$$

where $\mathbf{u}_k$ consists of the first q elements of $\mathbf{U}$. Next, defining the polynomials,

$$\begin{aligned}\mathbf{P}(z) &= \mathbf{K}_1 + \mathbf{K}_2 z^{-1} + \cdots + \mathbf{K}_n z^{n-1}\\ \mathbf{L}(z) &= \mathbf{K}_{n+1}z^{-1} + \mathbf{K}_{n+2}z^{-2} + \cdots + \mathbf{K}_{n+m-1}z^{m-1}\\ \mathbf{R}(z) &= \mathbf{R}_{N_p} + \mathbf{R}_{N_p-1}z^{-1} + \cdots + \mathbf{R}_1 z^{-N_p+1}\end{aligned}$$

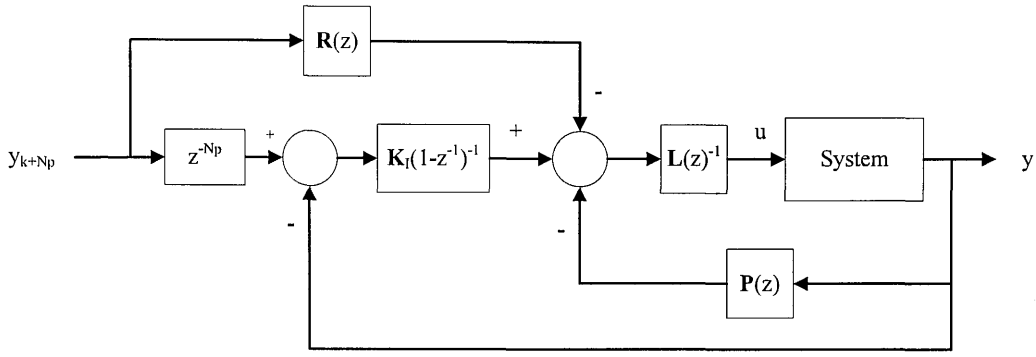

Figure 4.1: Block diagram of the MPC control scheme based on a model description with an integral–of–error state variable

and substituting $\mathbf{K}_{m+n}$ with $-\mathbf{K}_I$ and $\mathbf{z}_k = \frac{1}{1-z^{-1}}(\mathbf{r}_k - \mathbf{y}_k)$, the control law can be written in the following polynomial form:

$$\mathbf{u}_k = \mathbf{L}(z)^{-1}\left[\frac{1}{1-z^{-1}}\mathbf{K}_I(\mathbf{y}_k - \mathbf{r}_k) - \mathbf{P}(z)\mathbf{y}_k - \mathbf{R}(z)\mathbf{r}_{k+N_p}\right]$$

The unconstrained control system can now be represented by a block diagram as illustrated in Figure 4.1.

Defining the polynomials:

$$\mathbf{A}(z) = \mathbf{I}_p + \mathbf{A}_1 z^{-1} + \mathbf{A}_2 z^{-2} + \ldots + \mathbf{A}_n z^{-n}$$
$$\mathbf{B}(z) = \mathbf{B}_1 z^{-1} + \mathbf{B}_2 z^{-2} + \ldots + \mathbf{B}_m z^{-m}$$

that correspond to the left Matrix Fraction Description (MFD) form (Kailath, 1980) of the system input–output relationship (i.e. $\mathbf{y}_k = \mathbf{A}(z)^{-1}\mathbf{B}(z)\mathbf{u}_k$), straightforward manipulation of the block diagram in Figure 4.1 leads to the following closed–loop transfer function:

$$\mathbf{T}(z) = \left[\mathbf{A}(z)^{-1}\mathbf{B}(z)\mathbf{L}(z)^{-1}\left(\mathbf{K}_I - (1-z^{-1})\mathbf{P}(z)\right) - (1-z^{-1})\mathbf{I}_p\right]^{-1}$$
$$\left[\mathbf{A}(z)^{-1}\mathbf{B}(z)\mathbf{L}(z)^{-1}\left(\mathbf{K}_I + (1-z^{-1})\mathbf{R}(z)z^{-N_p}\right)\right]$$

for which $\lim_{z\to 1}\mathbf{T}(z) = \mathbf{I}_p$. This means that the closed–loop will follow any constant reference input with no steady state error. Similarly, the transfer function from a control input disturbance to the output is:

$$\mathbf{S}_i(z) = \left[(1-z^{-1})\mathbf{I}_p - \mathbf{A}(z)^{-1}\mathbf{B}(z)\mathbf{L}(z)^{-1}\left(\mathbf{K}_I - (1-z^{-1})\mathbf{P}(z)\right)\right]^{-1} \left[(1-z^{-1})\mathbf{A}(z)^{-1}\mathbf{B}(z)\right] \quad (4.20)$$

and from an output (load) disturbance to the output:

$$\mathbf{S}_o(z) = \left[(1-z^{-1})\mathbf{I}_p - \mathbf{A}(z)^{-1}\mathbf{B}(z)\mathbf{L}(z)^{-1}\left(\mathbf{K}_I - (1-z^{-1})\mathbf{P}(z)\right)\right]^{-1} \left[(1-z^{-1})\mathbf{I}_p\right] \quad (4.21)$$

In the above equation the subscripts $_i$ and $_o$ refer to the input and output disturbance transfer functions respectively. From equations (4.20) and (4.21) it follows that,

$$\begin{aligned}\lim_{z\to 1}\mathbf{S}_i(z) &= \mathbf{0}_{pq}\\ \lim_{z\to 1}\mathbf{S}_o(z) &= \mathbf{0}_p\end{aligned}$$

Hence, the closed–loop system rejects any constant input or output disturbances.

Remark 4.3.1. *At the steady state, if the desired output is in the feasible region (i.e. it can be reached with the available control inputs), the solution to the quadratic program will be the one of the unconstrained case. Therefore the steady state tracking and disturbance rejection properties hold for the constrained case as well.*

4.3.2 Stability analysis

As already mentioned in Section 2.4, a general stability analysis is performed in Appendix C. Since the constrained optimisation problem (4.12) is the same as

(C.1), the asymptotic stability of this alternative MPC formulation can be shown as in Appendix C.

4.4 Simulation Examples

In this section, two simulation examples are considered. The first example illustrates the reference following and disturbance rejection properties of the proposed control scheme, while the second example is an approach to a simulation benchmark and the complete design procedure is described. Furthermore, by this second example, an initial comparison is made between the proposed controller and the one of Wang and Young (2006).

4.4.1 Double integrating plant

Consider a continuous time double integrating plant (i.e. $G(s) = \frac{1}{s^2}$), that is a hard to control open loop unstable system. Using a sample time of $1s$, the discrete time transfer function is:

$$G(z^{-1}) = \frac{0.5z^{-1} + 0.5z^{-2}}{1 - 2z^{-1} + z^{-2}}$$

To represent the above transfer function in the state space form of (3.1), the following non–minimal state vector is selected,

$$\mathbf{x}_k^T = [y_k \quad y_{k-1} \quad u_{k-1} \quad z_k]$$

and the system matrices are given by,

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0.5 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -2 & 1 & -0.5 & 1 \end{bmatrix}; \mathbf{B} = \begin{bmatrix} 0.5 \\ 0 \\ 1 \\ -0.5 \end{bmatrix}; \mathbf{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

For the purposes of the present example, the prediction and control horizons are chosen as $N_p = 5$ and $N_c = 3$ respectively, that are relatively short but still lead to an acceptable response. By trial and error, the matrix $\mathbf{Q}$ is chosen as the 4×4 matrix with the value $\frac{5}{15}$ on it's diagonal, while $\mathbf{R}$ vanishes to a single constant that is chosen to be $\frac{1}{3}$. It is evident that they are both defined here as ratios between the *weight* and the prediction or control horizon. Clearly, other ways to choose the weighting matrices are possible, but the way it is done here reflects the relative weighting between the state and control action. As is presented in more detail in Chapter 5 an optimisation procedure can be followed to define the weighting matrices regarding the design objectives.

The control objectives are to keep the output of the plant at the origin, despite the presence of input and output disturbances, while also satisfying the saturating constraints on the input ($|u| \leq 10$ units) and rate–of–change ($|\Delta u| \leq 7$ units/s).

For the simulation presented in Figure 4.2, a step output disturbance of magnitude 15 units is applied to the system after 5 seconds from the start of the experiment, while a step input disturbance of -5 units is applied after 20 seconds. As expected, the controller rejected both the input and output disturbances, while ensuring type one servomechanism performance. Furthermore, both the input and rate–of–change are kept within the defined bounds.

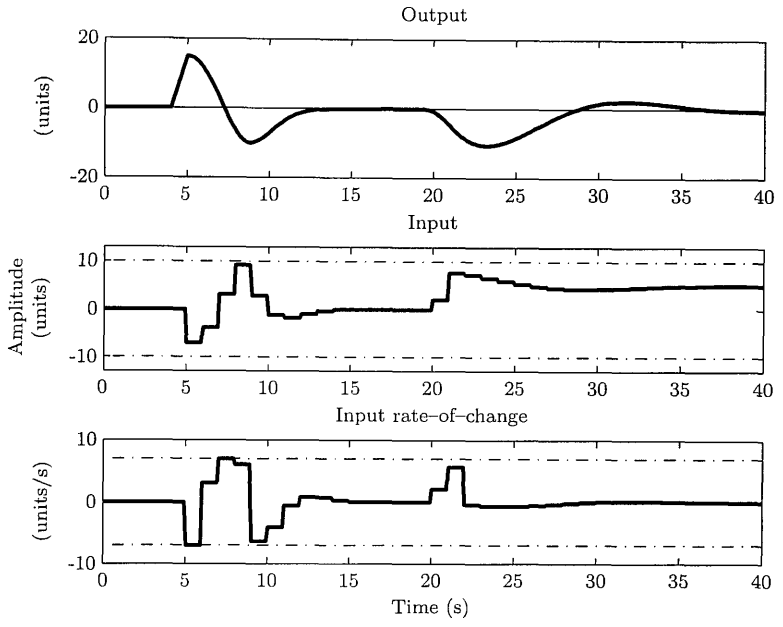

Figure 4.2: Presentation of the disturbance rejection properties with a closed loop simulation of the double integrating plant. Top subplot: reference level and output. Middle subplot: control input (solid trace) and constraints (dash-dot); Lower subplot: input rate–of–change (solid) and constraints (dash-dot).

4.4.2 The IFAC '93 benchmark

The IFAC '93 benchmark (Whidborne et al., 1995) is a difficult problem since it includes stringent closed loop performance requirements, despite the fact that the parameters of the plant are known only within a certain range. In fact, the plant operates at three different stress levels, with higher stress level implying greater time variations of the unknown parameters. The deterministic 7th order continuous time transfer function has the following form:

$$G_t(s) = \frac{K(-T_2 s + 1)\omega_0^2}{(s^2 + 2\zeta\omega_0 s + \omega_0^2)(T_1 s + 1)} \frac{\omega_\delta^2}{(s^2 + 2\zeta_\delta\omega_\delta s + \omega_\delta^2)(T_1^\delta s + 1)(T_2^\delta s + 1)} \quad (4.22)$$

where $T_1 = 5$, $T_2 = 0.4$, $\omega_0 = 5$, $\zeta = 0.3$, $K = 1$, $T_1^\delta = \frac{1}{8}$, $T_2^\delta = \frac{1}{12}$, $\omega_\delta = 15$ and $\zeta_\delta = 0.6$. For the closed–loop tests, the command level is specified as a square wave varying between $+1$ and -1 with a period of 20s, while an over/undershoot of 0.2 is allowed. In cases where these limits are exceeded, the absolute output

constraint of +1.5 and −1.5 should not be violated[2]. The rise time should be designed to be as fast as possible, without compromising the robustness. Finally, the plant input is constrained between the limits -5 and +5 and saturates at these levels.

To obtain the control model (2.1), the plant is simulated in open loop and data collected at a sampling rate of 0.5 seconds. Experimentation suggests that such a sampling rate allows for the estimation of a suitable backward shift operator model and is fast enough to handle the stochastic disturbances associated with the time varying parameters (see Taylor et al., 2001, for a complete discussion and application of the PIP controller to the same problem). As described in Section 2.1, the SRIV algorithm is used to estimate the model for this system. The discrete time transfer function model obtained in this manner takes the form:

$$H(z^{-1}) = \frac{0.0946z^{-2}}{1 - 0.9055z^{-1}} \tag{4.23}$$

The prediction and control horizons are $N_p = 50$ and $N_c = 10$ respectively, while the matrix $\mathbf{Q}$ is assigned the value $\frac{50}{50}$ on it's diagonal, and $\mathbf{R}$ is set to $\frac{1}{10}$ (chosen by trail and error to provide an acceptable closed loop response for the nominal case). The output and control signal when this controller is applied to the actual 7th order transfer function are depicted in Figure 4.3. This response resembles the one achieved by Taylor et al. (2001) (by visual inspection). However, since there is no detailed data (rise time, settling time, etc.) a direct comparison is not possible.

To assess the robustness of the design, the benchmark is subsequently simulated for the positive and negative extremes of Stress Level 2 of the system parameters as they are presented in Table 4.1. This results in 32 realisations that are depicted in Figure 4.4. This is merely to show that the proposed MPC design retains the robustness properties of the PIP realisation based on the same system (Taylor

[2]In the following, these are referred to as hard constraints. However it is not meant that they can not be violated, but rather that in an acceptable design they should not be violated.

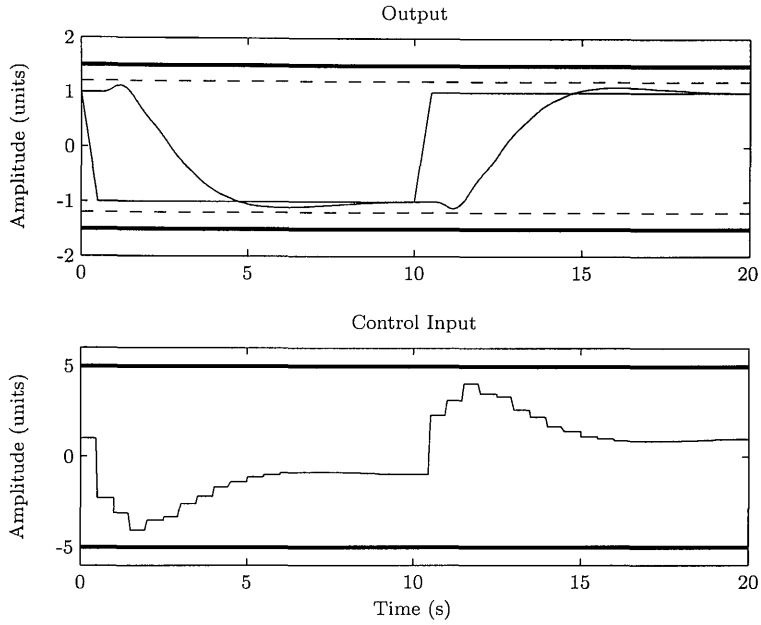

Figure 4.3: Nominal response for the IFAC '93 benchmark. Top subplot: reference level and output (thin solid lines) along with the soft (dashed lines) and hard (thick solid lines) output constraints; Lower subplot: Control input (thin solid line) and constraints (thick solid lines).

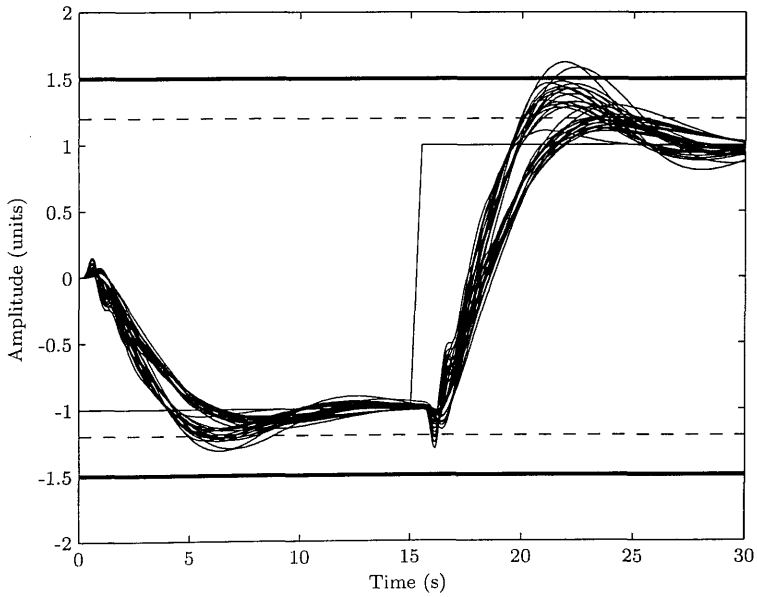

Figure 4.4: Robustness test for the IFAC '93 benchmark of the MPC controller based on a model with an integral–of–error state variable. The soft (dashed lines) and hard (thick solid lines) output constraints are also shown.

Table 4.1: Parameter variations for the 3 Stress Levels of the IFAC '93 Benchmark.

Stress Level	δT_1	δT_2	$\delta\omega_0$	$\delta\zeta$	δK
1	± 0.20	± 0.05	± 1.50	± 0.10	0
2	± 0.30	± 0.10	± 2.50	± 0.15	± 0.15
3	± 0.30	± 0.15	± 3.00	± 0.15	± 0.50

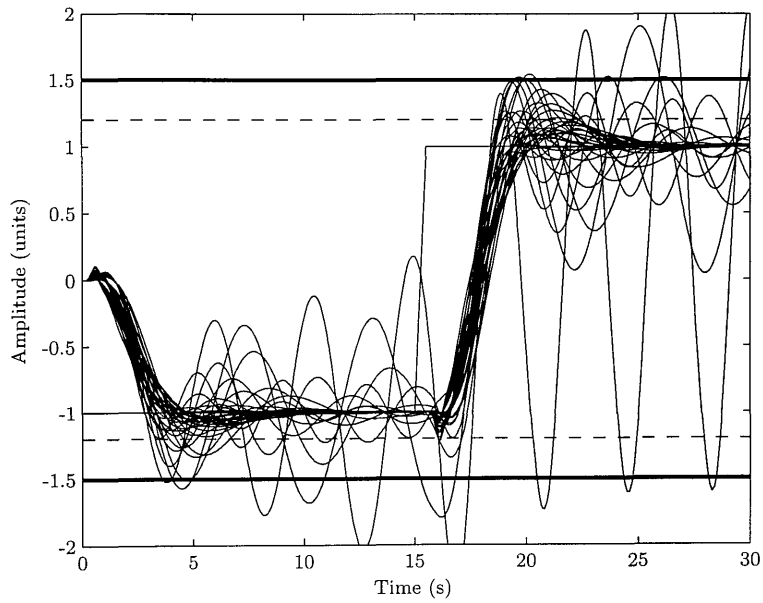

Figure 4.5: Robustness test for the IFAC '93 benchmark of the NMSS/MPC controller of Wang and Young (2006). The soft (dashed lines) and hard (thick solid lines) output constraints are also shown.

et al., 2001). It is clear however, that some of the realisations violate the output constraint that represents an unwanted system response. At this point reference is made again to the approach of Taylor et al. (2001) where at their initial control design the closed loop system also violated the output constraints when simulated for Stress Level 2. To overcome this problem, they optimised the weighting parameters of the PIP controller. This procedure is presented in Chapter 5 where application to MPC controllers is considered.

For comparative purposes, the same robustness test is performed for the MPC controller formulation of Wang and Young (2006) and the result is shown in Figure 4.5. In this regard, the NMSS/MPC controller of Wang and Young (2006) is tuned to produce a closed–loop response with a similar settling time as the one

proposed here, in the nominal case, so that the two controllers are comparable. However, as can be seen by direct comparison of Figures 4.4 and 4.5 they yield different responses when model mismatch is introduced. It is also evident that the proposed NMSS/MPC controller, utilising an explicit integral–of–error state variable, yields a higher proportion of well behaved responses than the equivalent controller of Wang and Young (2006). Of course, these figures represent just one simulation example and are merely presented to show that the controller formulation proposed here can in some cases produce better results than existing controllers.

Remark 4.4.1. *Although the results in this chapter have initiated from earlier research on the PIP framework, tuning of the proposed MPC scheme can present difficulties. It clearly resembles the infinite time LQ optimal approach that is used for PIP controllers, but the fact that the optimisation is performed over a finite horizon alters the effect of the weightings depending on the prediction and control horizons (an unacceptable initial tuning is later shown in Section 5.3.1). The fact however that the weightings refer to specific meaningful parts of the state vector (as in any NMSS model based on the approach of Young et al. (1987)) is evaluated in the next chapter to optimise them and is shown that the present formulation of MPC can provide the designer with more freedom when tuning the controller.*

4.5 Conclusion

This chapter is motivated by earlier research on MPC using a NMSS form (Wang and Young, 2006). In that work, the *differenced* values of the input and output signal are utilised in the non–minimal state vector, to ensure type one servomechanism performance. Such NMSS/MPC design has potential performance and robustness benefits, when compared to conventional MPC using a minimal state space model with observer as is shown by simulation examples in Wang and Young (2006).

The present chapter developed an alternative framework for NMSS/MPC design based on the directly measured (rather than differenced) values of the input and output variables, together with an *integral–of–error* state variable that is used to ensure type one servomechanism perfomance. This approach has close parallels with linear PIP methods, particularly with regard to the tuning of the control weighting matrices. However, the advantage of the NMSS/MPC approach is the inherent handling of constraints, albeit at an increased computational cost (by increase of the state vector and hence the optimisation problem).

In the simplest 'trial and error' case, the control engineer can adjust the total weights assigned to the input and output variables, together with the integral–of–error state, to achieve satisfactory performance. Such an approach has analogies with tuning the classical three term PID algorithm, albeit within a much more powerful framework. More advanced methods, such as multi–objective optimisation of the weighting matrices in $\mathbf{Q}$ and $\mathbf{R}$ are also possible (Tych et al., 1996; Chotai et al., 1998) and is considered in Chapter 5 for MPC controllers.

In both cases, the explicit integral–of–error state variable may provide additional degrees of freedom allowing for improved control. In fact, a full comparison of the proposed NMSS/MPC algorithm, the approach of Wang and Young (2006) and conventional *minimal* MPC methods, is the subject of Section 5.3 in Chapter 5. However, preliminary simulation results presented in this chapter have shown that the robustness and performance characteristics of the new approach are at least as good, or better, as those obtained by Wang and Young (2006). Finally, this chapter has shown by analysis and simulation, that the required set point tracking and disturbance rejection properties are maintained. Initial results of this alternative NMSS/MPC formulation have also been presented by the author in Exadaktylos et al. (2006).

Chapter 5

Multi–objective optimisation

Model Predictive Control (MPC) is a control technique that has been extensively used in the last decades (Morari, 1994). Most of the research conducted in the field has been towards increasing the robustness of the controllers (e.g. Kothare et al., 1996; Bemporad and Morari, 1999) and reducing the online computational load (e.g. Bemporad et al., 2002; Grieder et al., 2004).

In all the above cases, it is assumed that the cost function weighting matrices are process dependent and defined by economic or process related factors that are out of reach for the designer of the controller. However, it is possible to face a control problem that states some process related constraints (e.g. actuator constraints) and also defines control objectives that need to be achieved (e.g. rise time, overshoot, settling time). Especially in the Multi–Input Multi–Output (MIMO) case, it is sometimes of great importance to dynamically decouple the system outputs. In this regard, various approaches have been presented in the literature (e.g. Wang et al., 2000, 2002; Gilbert et al., 2003; Liu et al., 2007) for analytical decoupling of PI/PID controllers, an approach that is not applicable to MPC due to the need for online optimisation at every sampling instant and the presence of constraints.

In the direction of optimising the controller performance for specific design objectives, Evolutionary Algorithms (EAs) have recently been used in control prob-

lems (see Fleming and Purshouse, 2002, for a review). However, apart from the approaches of Vlachos et al. (1999) and Mei and Goodall (2000), they are mostly evaluated to deal with the control problem itself rather than tune existing control structures. In the direction of controller tuning, Fleming and Pashkevich (1986) have applied the goal attainment method to design a SISO compensator, while Tych et al. (1996) andChotai et al. (1998) evaluated the same method in order to tune PIP controllers for MIMO systems.

However, the problem of designing an MPC controller that achieves performance objectives has not been addressed in the relevant literature. Although the approach of Omnen et al. (1997) uses a Genetic Algorithm (GA) to solve the MPC problem, it is still in the category of solutions that calculate the control action rather than tune the controller.

This chapter presents a preliminary approach to performance optimisation of MPC controllers and addresses some of the issues that arise in the tuning process. In this regard, the goal attainment method that has already been used in the tuning of PIP controllers (e.g. Tych et al., 1996) and the design of a SISO compensator Fleming and Pashkevich (1986) is utilised in tuning MPC controllers. The approach is presented for the cases of minimal MPC design, the NMSS/MPC controller of Wang and Young (2006) and the alternative formulation that is based on a system with an explicit integral–of–error state that has been presented in Chapter 4. It is shown that this method can be used in tuning MPC controllers for both the SISO and MIMO cases, while in the latter, both performance (e.g. speed of response) and dynamic decoupling can be achieved. Furthermore it is suggested that the NMSS formulation presented in Chapter 4 provides the algorithm with extra freedom (because of the structure of the state vector) to result in improved performance.

In the following, Section 5.1 briefly summarises the control structures and presents the partitioning of the weighting matrices that are later used by the sim-

ulation examples. Section 5.2 describes the goal–attainment method that is used to tune the controllers and applies the approach to the MPC case. Then, Section 5.3 shows the results of the multi–objective optimisation process by simulation examples, while Section 5.4 concludes the chapter by summarising the results.

5.1 Predictive Control Descriptions

In this section, the control descriptions that are later considered are briefly presented. Although they have already been presented elsewhere in this thesis or are widely known in the relevant literature, the problem statement is presented here for completeness and a possible parametrisation of the weighting matrices is suggested.

5.1.1 MPC with an integral–of–error state variable

As already presented in Chapter 4 the NMSS/MPC can potentially provide some advantages compared to the controller of Wang and Young (2006). The cost function to be minimised at every sampling instant has initially been presented in (4.15a) and has the following form:

$$J_1 = \mathbf{X}^T \bar{\mathbf{Q}} \mathbf{X} + \mathbf{U}^T \bar{\mathbf{R}} \mathbf{U} \tag{5.1}$$

where the future state and control input vectors are given by (4.7a) and (4.13) respectively. For reasons that will be made clear in the following, the $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$

weighting matrices can be partitioned as:

$$\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{Q} & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \mathbf{Q} & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{Q} \end{bmatrix}, \bar{\mathbf{R}} = \begin{bmatrix} \mathbf{R} & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{R} & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \mathbf{R} & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{R} \end{bmatrix} \tag{5.2}$$

where $\mathbf{Q}$ is a square matrix of the dimension of the state vector and $\mathbf{R}$ is a square matrix with dimension the number of control inputs.

5.1.2 The controller of Wang and Young (2006)

In the same manner as before, the cost function to be minimised at each sampling instant (already given in (4.10)) has the form:

$$J_2 = (\mathbf{S} - \mathbf{Y})^T \bar{\mathbf{Q}} (\mathbf{S} - \mathbf{Y}) + \Delta \mathbf{U}^T \bar{\mathbf{R}} \Delta \mathbf{U} \tag{5.3}$$

where the future set point, the predicted output and control input increment vectors are given by (4.7b), (4.7c) and (4.9). Since the cost function (5.3) is based on the error of the output and not the state vector, the weighting parameters are in general less than in the case where the cost function is given by (5.1). In this regard, in a similar manner to (5.2), the weighting matrices $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ are parametrised in such a way that allows for weights to vary throughout the prediction horizon. That is:

$$\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q}_1 & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{Q}_2 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \mathbf{Q}_{N_p-1} & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{Q}_{N_p} \end{bmatrix}, \bar{\mathbf{R}} = \begin{bmatrix} \mathbf{R}_1 & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{R}_2 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \mathbf{R}_{N_c-1} & 0 \\ 0 & 0 & \cdots & 0 & \mathbf{R}_{N_c} \end{bmatrix} \tag{5.4}$$

Clearly the same partitioning of the $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ matrices can be adopted in the case of the controller of Section 5.1.1 that would provide extra design freedom. However, this is not considered necessary, at least in the examples considered here, since the form described in (5.2) can already produce the desired results.

5.1.3 The minimal controller

In the case of an MPC controller based on a minimal state space representation, the cost function can take various forms, depending on the problem formulation. In the examples presented here, to account for type one servomechanism performance without the need for coordinate transformation, a cost identical to the one of Wang and Young (2006) is considered (i.e. the cost function is the one depicted by (5.3)). A cost similar to (5.1) is possible but, in the general case, this would require a coordinate transformation to ensure steady state output tracking that is inherent in the NMSS description used by the approach described in Chapter 4. In the following, the weighting matrices $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ for the case of the minimal MPC controller are also parametrised as in (5.4).

5.2 Tuning the MPC controllers

It is evident from the above that the choice of $\mathbf{Q}$ and $\mathbf{R}$ affects the controller behaviour and therefore the closed loop system response. Although these weights are sometimes defined by the process itself, or are dealt with by intuition, this does not always result in the desired system behaviour. Even for the SISO case, where a fairly good understanding of the input–output correlation can be achieved, it is not always easy for all of the design objectives (rise time, overshoot, settling time, etc.) to be achieved. Furthermore, in more complex MIMO systems with strong cross coupling among the inputs and the outputs, objectives such as dynamic output decoupling are difficult to achieve by intuitively choosing the cost function weights.

In this regard, Tych et al. (1996) evaluated goal attainment to optimise the

LQ cost function weights to meet multiple design objectives in the case of the PIP controller, while Mei and Goodall (2000) used GAs for the same purpose in the case of an optimal H_2 controller. In the following, the goal attainment method is used for the optimal tuning of MPC controllers and guidelines are provided for choosing the weights to be optimised. For completeness, some background on the goal attainment method is also provided.

5.2.1 The goal attainment method

The goal attainment method involves expressing a set of goal objectives $\mathcal{F}^* = \{F_1^*, F_2^*, \ldots, F_n^*\}$, which are associated with a set of design objectives $\mathcal{F}(\mathbf{f}) = \{F_1(\mathbf{f}), F_2(\mathbf{f}), \ldots, F_n(\mathbf{f})\}$, where $\mathbf{f}$ is a vector of optimisation parameters. The problem formulation allows for the objectives to be over– or underachieved, allowing for very optimistic goals to be defined without leading to an infeasible problem. The relative weighting among the objectives can also define which of them are more important, allowing the des gner to define every desirable objective but over–evaluate those that necessarily need to be satisfied.

Defining the goal, objective and weighting vectors as,

$$\mathbf{F}^* = \begin{bmatrix} F_1^* & F_2^* & \cdots & F_n^* \end{bmatrix}^T$$

$$\mathbf{F}(\mathbf{f}) = \begin{bmatrix} F_1(\mathbf{f}) & F_2(\mathbf{f}) & \cdots & F_n(\mathbf{f}) \end{bmatrix}^T$$

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_n \end{bmatrix}^T$$

the optimisation problem can be defined as,

$$\min_{\lambda \in \mathbb{R}, \mathbf{f} \in \Omega} \lambda \tag{5.5}$$

such that

$$\mathbf{F}(\mathbf{f}) - \mathbf{w}\lambda \leq \mathbf{F}^*$$

In cases where a w_i is zero in the weighting vector, the respective objective is meant to be a hard one that will result in an optimisation vector that will satisfy it (if possible). Depending on the meaning of the optimisation parameters $\mathbf{f}$, it is sometimes necessary to constrain them to a specified region (linear or nonlinear). This is also possible in the context of goal attainment (region Ω in (5.5)). The examples here utilise the fgoalattain function from the optimisation toolbox in Matlab® to solve the goal attainment problem.

5.2.2 Application to MPC

To apply the optimisation method presented in the previous section to MPC, the optimisation parameters need to be identified. Any combination of the elements of the weighting matrices can be chosen with the additional constraint that the weighting matrices $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ need to be positive and semi–positive definite respectively. To account for this extra constraint, the Cholesky factorisation (e.g. $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$ where $\mathbf{L}$ is lower diagonal[1]) of the matrices can be used and the elements of $\mathbf{L}$ can be the parameters that need to be optimised (as in Chotai et al., 1998).

Following directions for tuning PIP controllers, general guidance is provided for tuning the MPC controller that is based on a system description with an integral–of–error state variable. In this regard, and to simplify the optimisation problem, a subset of the $\mathbf{Q}$ and $\mathbf{R}$ can be chosen to be optimised as follows:

$$\mathbf{Q} = diag\begin{bmatrix}\mathbf{w_y} & \cdots & \mathbf{w_y} & \mathbf{w_{u^-}} & \cdots & \mathbf{w_{u^-}} & \mathbf{w_z}\end{bmatrix} \tag{5.6a}$$

$$\mathbf{R} = \mathbf{w_u} \tag{5.6b}$$

$\mathbf{w_y}$, $\mathbf{w_{u^-}}$ and $\mathbf{w_z}$ correspond to weightings for the present and previous values of the outputs, the previous values of the inputs and the integral–of–error states of

[1]In case more elements of the $\bar{\mathbf{Q}}$ need to be optimised the Cholesky factorisation should of course be applied to the $\bar{\mathbf{Q}}$ and not to the $\mathbf{Q}$ (although positive definiteness of $\bar{\mathbf{Q}}$ implies that of $\mathbf{Q}$)

the state vector; while $\mathbf{w_u}$ corresponds to weightings for the current inputs. In any case the weightings $\mathbf{w}$ are square matrices of appropriate dimensions.

In both the controller of Wang and Young (2006) and the minimal case, the cost function for set–point following consists of the output error and the control action (or it's increment), the $\mathbf{Q}$ matrix in that case can have the form:

$$\mathbf{Q} = \mathbf{w_y}$$

while the $\mathbf{R}$ has the same form as before. An alternative to that, and because the number of optimisation parameters is limited, a further extension to the above can be made, allowing for both $\mathbf{Q}$ and $\mathbf{R}$ matrices to vary over the prediction and control horizons respectively. That is:

$$\begin{aligned} \mathbf{Q}_i &= \mathbf{w}_\mathbf{y}^{(i)} \quad , i = 1, \ldots, N_p \\ \mathbf{R}_i &= \mathbf{w}_\mathbf{u}^{(i)} \quad , i = 1, \ldots, N_c \end{aligned}$$

Any other combination of optimisation parameters is possible and mainly depends on the designer, the number of design objectives and the system to be controlled, but the above offers an adequate framework for performance optimisation as is shown in the next section by simulation examples.

5.3 Simulation Examples

This section presents the effectiveness of the proposed tuning method by application to simulation examples. First the SISO system of the IFAC '93 Benchmark is simulated by application of the controller of Chapter 4 and the controller of Wang and Young (2006) to demonstrate the applicability of the approach in cases with a relatively small number of optimisation parameters. Then the 'Shell Heavy Oil fractionator' simulation benchmark is utilised as a basis to exploit the dynamic

decoupling capabilities of the tuning technique and to compare the responses of the NMSS/MPC control structures and the minimal realisations of MPC presented in Section 5.1.

5.3.1 SISO case: The IFAC '93 benchmark

As already mentioned in Section 4.4.2, the IFAC '93 benchmark (Whidborne et al., 1995) is represented by the 7th order continuous–time transfer function (4.22) that needs to follow a square wave with lower value -1 units, higher value 1 units and period $T = 20$s. Furthermore, the control input is constrained to values between -5 units and 5 units, while the output should have as fast a rise time as possible and not exceed -1.5 units and 1.5 units (preferably -1.2 units and 1.2 units).

In the same manner as in Section 4.4.2 the necessary open loop simulations are performed to obtain a reduced order discrete time transfer function model. For MPC control based on a state space description with an explicit integral–of–error state, the above system is represented in the state space form (3.1), with the state vector:

$$\mathbf{x}_k = \begin{bmatrix} y_k & u_{k-1} & z_k \end{bmatrix}^T$$

For the present simulations, the prediction and control horizons are chosen to be $N_p = 10$ and $N_c = 5$ respectively. The initial weighting matrices are chosen as $\mathbf{Q} = diag[1 \quad 1 \quad 1]$ and $\mathbf{R} = 1$ in (5.2) (the reason for not choosing the same values as in Section 4.4.2 is explained in the following). The response for these values of the weights and horizons is shown as dashed lines in Figure 5.1. Evidently, this response does not satisfy the control objectives. Although in Section 4.4.2 an acceptable response was achieved (Figure 4.3) with a fairly intuitive choice of the weightings, this was mainly due to the long prediction and control horizons and the fact the the weighting for the control signal was chosen much smaller than the one for the state vector. In this simulation, the horizons are deliberately chosen shorter to depict the fact that in this case tuning the controller is not intuitively

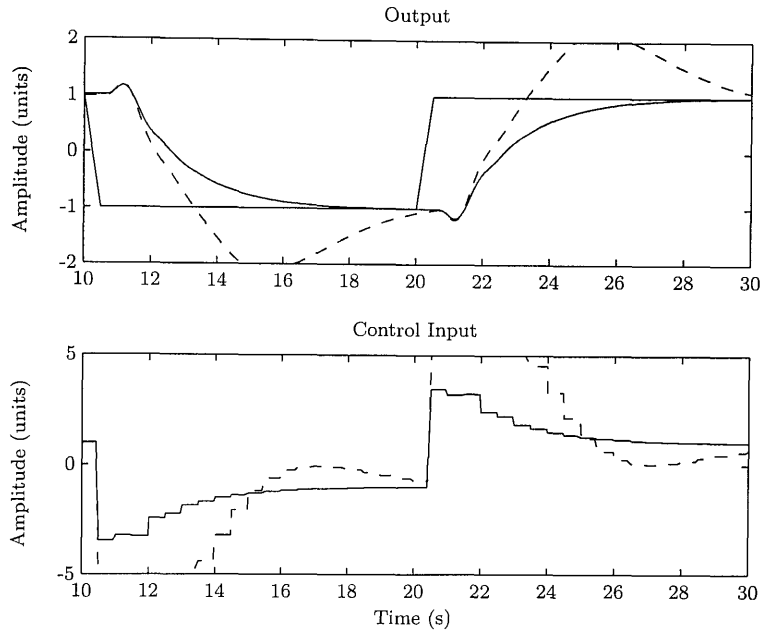

Figure 5.1: System output (top figure) and control signal (bottom figure) for the nominal case of the IFAC 93 benchmark simulation before (dashed lines) and after (solid lines) multi–objective optimisation of the weighting parameters for the controller based on a system with an explicit integral–of–error state variable.

straightforward.

Therefore, to improve the system response, the goal attainment methodology that is described in Section 5.2.1 is employed. In this regard, minimisation of the overshoot is chosen as the first objective and is considered that no overshoot is desired (i.e. $F_1^* = 1$). However, it is clear that the optimisation procedure can result in a very slow response without any overshoot. To avoid such a situation, the minimisation of the rise time is chosen as the second objective and is set to equal the rise time of the non–optimised response (i.e. $F_2^* = F_2(\mathbf{f}_0)$, where $\mathbf{f}_0$ is the vector of the initial values of the optimisation parameters). Such an objective is over–optimistic (and contradicting to the first objective) but as already mentioned it is possible to set very ambitious objectives without any practical implications. Finally, and to force a smooth system response, the integral of absolute error from the output to the set–point (i.e. $\int (|r(t) - y(t)|)\, dt$) is chosen to be minimised and the desired value is set to 80% that of the non–optimised

response (i.e. $F_3^* = 0.8F_3(\mathbf{f}_0)$).

Subsequently, and according to the general guidance provided in Section 5.2.2, the $\mathbf{Q}$ matrix is partitioned as:

$$\mathbf{Q} = diag\begin{bmatrix} w_y & w_{u^-} & w_z \end{bmatrix}$$

while $\mathbf{R} = w_u$. The optimisation parameter vector is then defined as:

$$\mathbf{f} = \begin{bmatrix} w_y & w_{u^-} & w_z & w_u \end{bmatrix}^T$$

After the optimisation procedure, the closed loop system clearly yields an improved response that is depicted with solid lines in Figure 5.1, and corresponds to the following weightings:

$$\begin{aligned} \mathbf{Q} &= diag\begin{bmatrix} 12.4315 & 1.3277 & 0.1222 \end{bmatrix} \\ \mathbf{R} &= 1.0155 \end{aligned}$$

For comparative purposes, and to show that the proposed technique can be directly applied to any MPC structure, the same procedure is followed to tune the MPC/NMSS controller of Wang and Young (2006). In this regard, the system is brought in the form of (4.5) with the following state vector:

$$\mathbf{x}_k = \begin{bmatrix} \Delta y_k & \Delta u_{k-1} & y_k \end{bmatrix}^T$$

The prediction and control horizons are chosen as before to be $N_p = 10$ and $N_c = 5$ respectively, while the initial weightings are set to $\mathbf{Q}_i = 1$ and $\mathbf{R}_j = 1$, for $i = 1, \ldots, N_p$ and $j = 1, \ldots, N_c$ respectively. This results in the system response depicted on Figure 5.2 in dashed lines. Although this is a fairly good response that satisfies the control objectives), it still has a quite long settling time and a slight overshoot. Subsequently it is shown that it can further be improved by evaluating

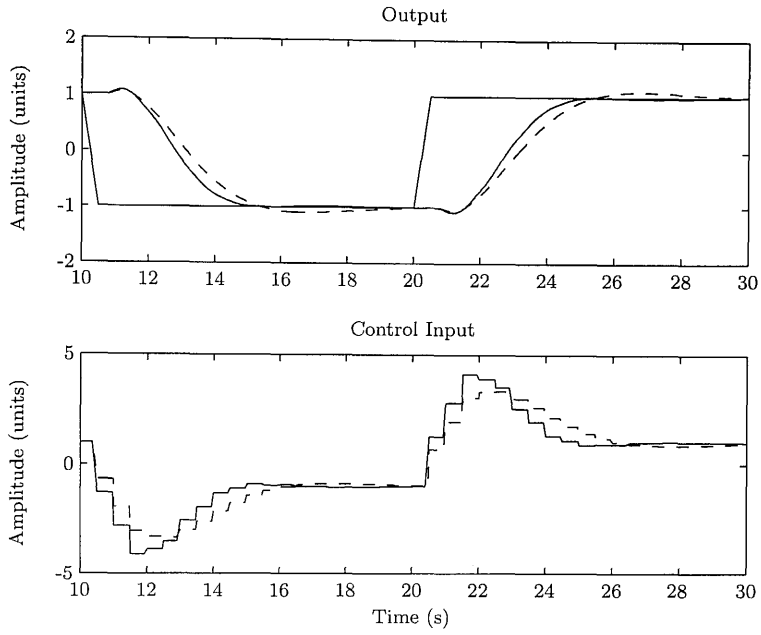

Figure 5.2: System output (top figure) and control signal (bottom figure) for the nominal case of the IFAC 93 benchmark simulation before (dotted lines) and after (solid lines) multi–objective optimisation of the weighting parameters for the controller of Wang and Young (2006)

the multi–objective optimisation procedure. In this regard, the same optimisation objectives as before are chosen and the $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ partitioning described in (5.4) is adopted. The improved response is shown as solid lines on Figure 5.2 and corresponds to the following weighting matrices:

$$\bar{\mathbf{Q}} = diag[1 \quad 1.0275 \quad 1.0395 \quad 1.0392 \quad 1.0317 \quad 1.022 \quad 1.0145 \quad 1.0085 \quad 1.004 \quad 1.0004]$$

and

$$\bar{\mathbf{R}} = diag\begin{bmatrix} 0.643 & 1.0935 & 1.0496 & 1.0233 & 1.0095 \end{bmatrix}$$

As can be seen from the values of the weighting matrices, in order for the improved performance to be achieved, the weights need to be slightly changed. However, this small change is difficult to be found without a procedure like the one presented above. This leads to the conclusion that the multi–objective performance optimisa-

Table 5.1: Steady State Gain (SSG), Time Constant (TC) and Time Delay (TD) for each input–output pathway for the Shell Heavy Oil Fractionator simulation

	Input 1			Input 2			Input 3		
	SSG	TC	TD	SSG	TC	TD	SSG	TC	TD
Output 1	4.05	50	27	1.77	60	28	5.88	45	27
Output 2	5.39	50	18	5.72	60	14	6.90	40	15
Output 3	3.66	9	2	1.65	30	20	5.53	40	2

tion can form a powerful framework in tuning MPC controllers. The next example presents a complete optimisation procedure of the performance of a MIMO plant while an example of optimisation for robust performance is given in Section 6.3.3.

5.3.2 MIMO case: The Shell Heavy Oil Fractionator Simulation

In this section, a MIMO example is considered, namely the 'Shell Heavy Oil Fractionator problem' (see Sandoz et al., 2000, for a brief description). This was introduced by Shell in 1987 and although it does not relate to any real system, it contains all the significant elements of a fractionator and can be used as a benchmark for multivariable control. However, in this case, it is evaluated as a basis to highlight the tuning properties presented in Section 5.2 and the control objectives defined by the benchmark are not considered (see Vlachos et al., 2002, for an approach to the actual benchmark simulation and it's objectives).

For completeness, a brief description of the benchmark is presented in the following. The fractionator consists of 5 input and 7 output variables (i.e. 35 input–output pathways with various steady state gains, time constants and time delays). However, only 3 of the outputs need to be controlled, while 2 of the inputs are uncontrollable disturbance signals. This leaves a 3–Input 3–Output system with the continuous time steady state gains, time constants and time delays of each input–output pathway shown in Table 5.1. To illustrate the format of Table 5.1, the relationship of the third input to the second output in continuous transfer

function from is:

$$y_2(s) = \frac{6.9e^{-40s}}{15s+1}u_3(s)$$

For convenience, the units employed here are seconds, although it is possible to use any other units that reflect the response of the plant under consideration. Furthermore, it should be noted that although the initial benchmark description requires the control signals to be constrained, this is not taken into account in the present simulations since the main goal here is to highlight the tuning technique and not to present a complete solution to the benchmark.

As already shown in previous examples, to form the MPC problem, open loop simulations are performed to the continuous plant and data are collected with a sampling rate of 10s. Subsequently, the SRIV algorithm is applied for model order detection and system identification as is discussed in Section 2.1. Four Multi Input Single Output systems are identified and combined in one MIMO system described by the following difference equation:

$$\mathbf{y}_k + \mathbf{A}_1\mathbf{y}_{k-1} + \mathbf{A}_2\mathbf{y}_{k-2} + \mathbf{A}_3\mathbf{y}_{k-3} = \mathbf{B}_1\mathbf{u}_{k-1} + \mathbf{B}_2\mathbf{u}_{k-2} + \mathbf{B}_3\mathbf{u}_{k-3} + \mathbf{B}_4\mathbf{u}_{k-4} + \mathbf{B}_5\mathbf{u}_{k-5} \tag{5.7}$$

in which:

$$A_1 = \begin{bmatrix} -1.6637 & 0 & 0 \\ 0 & -2.4418 & 0 \\ 0 & 0 & -1.9553 \end{bmatrix}; \quad A_2 = \begin{bmatrix} 0.6918 & 0 & 0 \\ 0 & 1.9864 & 0 \\ 0 & 0 & 1.2334 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -0.5383 & 0 \\ 0 & 0 & -0.2473 \end{bmatrix}$$

and

$$B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.4173 & 0.5851 \\ 0 & 0 & 3.9167 \end{bmatrix}; \qquad B_2 = \begin{bmatrix} 0.1232 & 0 & 0.1789 \\ 0.9853 & -0.2036 & -0.0301 \\ 0.9993 & 0.8873 & -6.0118 \end{bmatrix}$$

$$B_3 = \begin{bmatrix} 0.5079 & 0.2737 & 0.7375 \\ -1.6010 & -0.4722 & -1.1641 \\ -1.0690 & -0.6085 & 2.3036 \end{bmatrix}; \qquad B_4 = \begin{bmatrix} -0.5173 & -0.2241 & -0.7511 \\ 0.6493 & 0.2940 & 0.6520 \\ 0.1557 & -0.2109 & 0 \end{bmatrix}$$

$$B_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.0492 & 0.1303 & 0 \end{bmatrix}$$

5.3.2.1 Dynamic Decoupling

From this point on, the difference equation (5.7) is considered and no mention is made to the actual continuous time plant. This is mainly to emphasise the tuning approach, it's capabilities and the differences among different MPC structures (namely the NMSS controller described in Chapter 4, the one of Wang and Young (2006) and the minimal MPC realisation as described in Section 5.1.3).

In this regard, the above system is expressed in the state space form of (3.1) for application of the NMSS controller presented in Chapter 4, with the state vector:

$$\mathbf{x}_k^T = \begin{bmatrix} \mathbf{y}_k^T & \mathbf{y}_{k-1}^T & \mathbf{y}_{k-2}^T & \mathbf{u}_{k-1}^T & \mathbf{u}_{k-2}^T & \mathbf{u}_{k-3}^T & \mathbf{u}_{k-4}^T & \mathbf{z}_k^T \end{bmatrix}$$

where,

$$\mathbf{y}_k = \begin{bmatrix} y_{1,k} & y_{2,k} & y_{3,k} \end{bmatrix}^T$$

$$\mathbf{u}_k = \begin{bmatrix} u_{1,k} & u_{2,k} & u_{3,k} \end{bmatrix}^T$$

For the purposes of the present example, the prediction and control horizons are

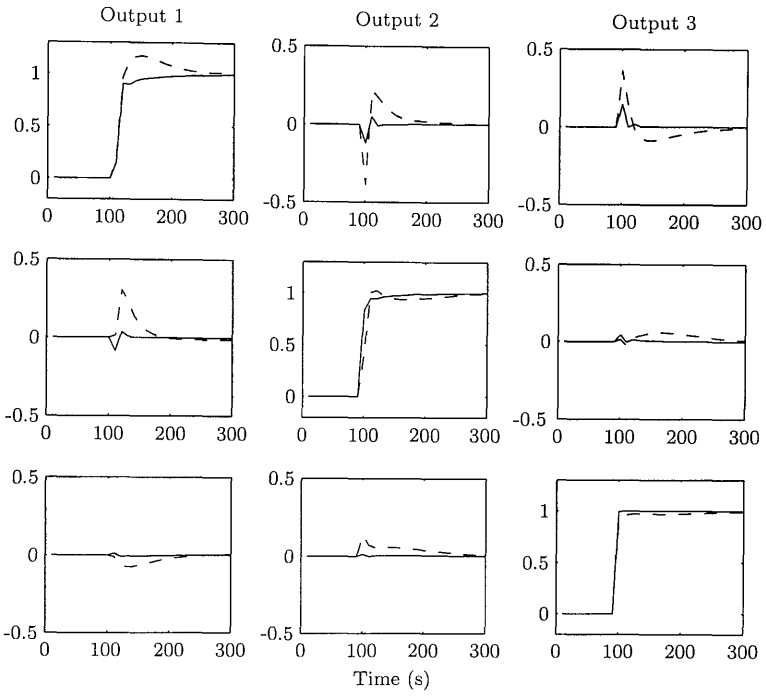

Figure 5.3: System outputs for the Shell Heavy Oil fractionator example. Each line corresponds to a step to the reference of the respective output. In every case the controller based on a system description with an explicit integral–of–error state is employed before (dashed lines) and after (solid lines) multi–objective optimisation

chosen $N_p = 10$ and $N_c = 5$ respectively, while the $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ matrices are chosen as the identity matrices of appropriate dimensions (for this non–optimised simulation). Applying a step change of amplitude 1 unit to the reference of each output, results to the system response that is depicted with dashed lines in Figure 5.3. Here, each line of figures corresponds to a step reference change to the respective output, while the off–diagonal plots show the cross–coupling among the outputs. From Figure 5.3 the cross–coupling is apparent and since in many cases such a response in not acceptable, it is shown in the following that it can be reduced and in some cases even be eliminated. In this regard, the partitioning depicted in (5.6) is adopted for the $\mathbf{Q}$ and $\mathbf{R}$ matrices while the $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ have the form (5.2). The optimisation problem has 9 parameters that need to be minimised (i.e. the integral of absolute error between each output and the corresponding reference level in all three simulations) and 15 optimisation parameters that result from the

parameterisation of $\mathbf{w_y}$, $\mathbf{w_{u^-}}$, $\mathbf{w_u}$ and $\mathbf{w_z}$, were the first three are 3×3 diagonal matrices (i.e. 3 parameters for every matrix) while the last one is a symmetric positive definite 3×3 matrix (i.e. 6 parameters). After optimisation[2], the weighting matrices are:

$$\mathbf{w_y} = diag \begin{bmatrix} 0.056 & 1 & 0.629 \end{bmatrix}$$

$$\mathbf{w_{u^-}} = diag \begin{bmatrix} 0.01 & 0.01 & 0.01 \end{bmatrix}$$

$$\mathbf{w_u} = diag \begin{bmatrix} 0.01 & 0.01 & 0.1163 \end{bmatrix}$$

and

$$\mathbf{w_z} = \begin{bmatrix} 0.0360 & 0.0050 & 0.0050 \\ 0.0050 & 0.0050 & 0.2118 \\ 0.0050 & 0.2118 & 0.1923 \end{bmatrix}$$

The optimised responses are shown in Figure 5.3 as solid lines, and it can be seen that the cross–coupling has been considerably reduced without substantially affecting the response of the output that needs to follow the step. As already stated, any combination of the weighting matrices can be chosen as the parameters to be optimised that can lead to different results. The above is chosen since previous experience on PIP control suggests that only the diagonal elements of the weighting matrices for the present and past input and output variables are optimised, while the diagonal and off–diagonal elements of the weighting that correspond to the integral–of–error variables are optimised.

The same control design procedure is followed for the controller of Wang and Young (2006), were the system is expressed by the form (4.5) with the following

[2]The values within the matrices are deliberately constrained to a small positive value to avoid conditioning problems in the optimisation procedure. It can also be seen that many of the values of the weighting matrices are actually on this limit.

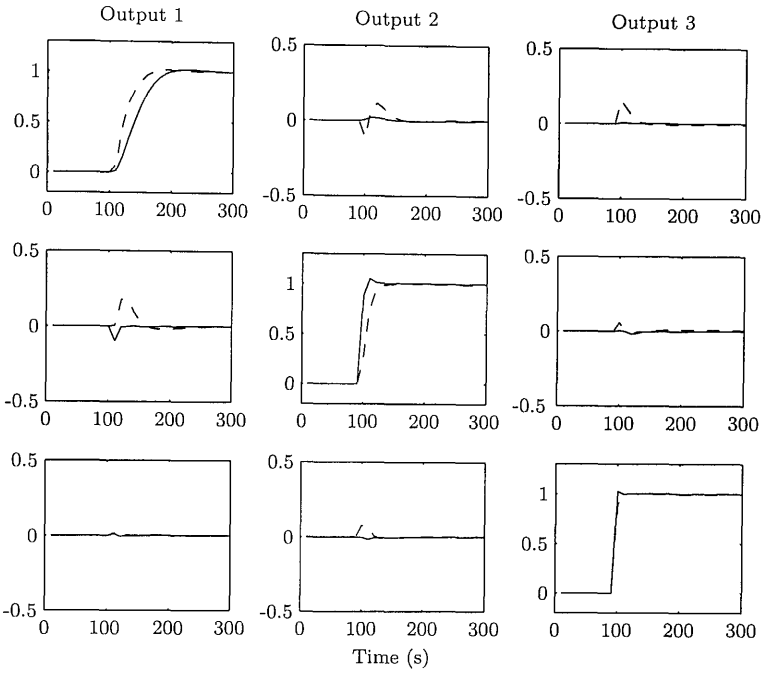

Figure 5.4: System outputs for the Shell Heavy Oil fractionator example. Each line corresponds to a step to the reference of the respective output. In every case the controller of Wang and Young (2006) is employed before (dashed lines) and after (solid lines) multi–objective optimisation

state vector:

$$\mathbf{x}_k^T = \left[\Delta\mathbf{y}_k^T \quad \Delta\mathbf{y}_{k-1}^T \quad \Delta\mathbf{y}_{k-2}^T \quad \Delta\mathbf{u}_{k-1}^T \quad \Delta\mathbf{u}_{k-2}^T \quad \Delta\mathbf{u}_{k-3}^T \quad \Delta\mathbf{u}_{k-4}^T \quad \mathbf{y}_k^T\right]$$

By choosing the prediction and control horizons $N_p = 10$ and $N_c = 5$ respectively and the weighting matrices $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ to be the identity matrices, the resulting responses are shown in Figure 5.4 as dashed lines. Although the cross–coupling is not as evident[3] as in the case of the controller with an explicit integral–of–error state (which is the result of the initial choice of the weighting matrices), the same optimisation procedure is evaluated in order to reduce it even further (whilst trying to preserve the system response to the step input). In this regard, the parametri-

[3]As can be seen in the top left sub–plot of Figure 5.4, the step response of the first output is much slower than the respective one in Figure 5.3 which is probably the reason for the reduced cross–coupling of the outputs. Reducing the speed of response in the controller with an explicit integral–of–error state will probably decrease the cross–coupling.

sation (5.4) is chosen for $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$, while the $\mathbf{Q}_i$ and $\mathbf{R}_j$ matrices are chosen to be equal for $i = 1, \ldots, N_p$ and $j = 1, \ldots, N_c$ respectively. Furthermore, they are required to be positive definite, hence the Cholesky factorisation introduced in Section 5.2.2 is used. The resulting optimised responses where the cross–coupling is almost eliminated are shown as solid lines in Figure 5.4, that correspond to the following weighting matrices:

$$\mathbf{Q} = \begin{bmatrix} 0.1000 & 0.1572 & 0.05 \\ 0.1572 & 6.5895 & 0.05 \\ 0.0500 & 0.0500 & 10 \end{bmatrix}; \qquad \mathbf{R} = \begin{bmatrix} 2.2165 & 0.7181 & 3.4158 \\ 0.7181 & 0.2831 & 1.3949 \\ 3.4158 & 1.3949 & 1.6922 \end{bmatrix}$$

5.3.2.2 'Designed For' Response

This section presents a more realistic and demanding design example, where not only dynamic decoupling of the outputs is required, but also a specific system response is desirable. In this regard, each of the system outputs is required to have a step response that is as close as possible to a first order system with one real pole located at 0.7 in the z–plane (and a steady state gain of unity) as shown by the transfer function (5.8) while the cross coupling should be reduced to a minimum.

$$h(z^{-1}) = \frac{0.3}{1 - 0.7z^{-1}} \tag{5.8}$$

The same procedure as before is followed and the results are presented in Figure 5.5.

In more detail, each row of Figure 5.5 corresponds to the optimised response for a different control structure (the first row is for the controller based on a system description with an explicit integral–of–error state, the middle row for the controller of Wang and Young (2006) and the bottom row for the minimal MPC controller as described in Section 5.1.3). Furthermore, each column represents a step to the respective input reference signal. For example, the sub–plot with coordinates $(2, 3)$ corresponds to a step reference change in the third output for

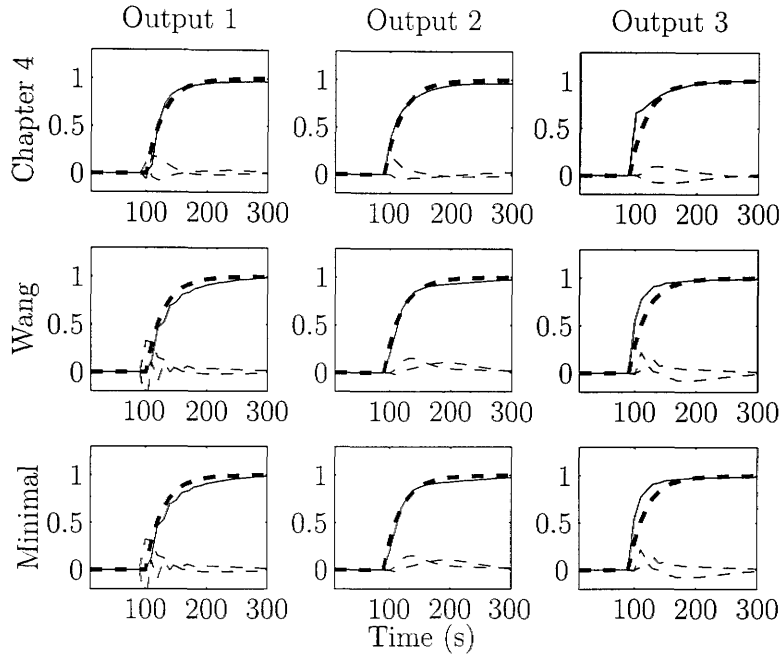

Figure 5.5: 'Designed for' response of the Shell Heavy Oil fractionator example for the controller presented in Chapter 4 (top row), the controller of Wang and Young (2006) (middle row) and a minimal MPC controller (bottom row). Each column corresponds to a step to the reference of the respective output. In every case the desired output is the thick dashed line, the respective output is the thin solid line and the outputs that should remain to zero are plotted as thin dashed lines.

the controller of Wang and Young (2006). There, the desired response should be as close to the one presented by the thick dashed line (the actual is shown as a solid line), while the responses of the first and second outputs (dashed lines) should remain as close to zero as possible.

To obtain the above results, the weighting matrices for the controller presented in Chapter 4 are partitioned as in Section 5.2.2 with the respective matrices being:

$$\mathbf{w_y} = \begin{bmatrix} 0.9873 & 0.1308 & 0.0097 \\ 0.1308 & 0.4687 & 0.0433 \\ 0.0097 & 0.0433 & 0.8913 \end{bmatrix}; \qquad \mathbf{w_{u^-}} = \begin{bmatrix} 0.4785 & 0.5000 & 0.5000 \\ 0.5000 & 0.3093 & 0.0050 \\ 0.5000 & 0.0050 & 0.1895 \end{bmatrix}$$

and

$$\mathbf{w_z} = \begin{bmatrix} 0.0792 & 0.0167 & 0.0050 \\ 0.0167 & 0.0327 & 0.0140 \\ 0.0050 & 0.0140 & 0.0758 \end{bmatrix}; \qquad \mathbf{w_u} = \begin{bmatrix} 0.9705 & 0.0648 & 0.4984 \\ 0.0648 & 0.9987 & 0.0050 \\ 0.4984 & 0.0050 & 0.0154 \end{bmatrix}$$

The optimised weighting matrices for the controller of Wang and Young (2006) are chosen to be constant throughout the prediction and control horizons and their values are:

$$\mathbf{Q}_{wang} = \begin{bmatrix} 0.0642 & 0.0050 & 0.0612 \\ 0.0050 & 0.2921 & 0.0050 \\ 0.0612 & 0.0500 & 0.2344 \end{bmatrix}; \qquad \mathbf{R}_{wang} = \begin{bmatrix} 1.8544 & 1.7499 & 0.6671 \\ 1.7499 & 1.6666 & 1.0602 \\ 0.6671 & 1.0602 & 6.1138 \end{bmatrix}$$

To obtain a minimal representation for the design of a minimal MPC controller, the obsvf function of the MATLAB® Control System toolbox is evaluated that computes the observability staircase form of a state space system and the observable part is used. The state observer is set to have poles close to the origin (random real poles with zero mean and 0.1 standard deviation), so that it has a near dead–beat response[4]. The optimisation procedure is subsequently evaluated that results in the following weighting matrices (constant throughout the prediction and control horizons):

$$\mathbf{Q}_{minimal} = \begin{bmatrix} 0.0664 & 0.0050 & 0.0634 \\ 0.0050 & 0.3033 & 0.0050 \\ 0.0634 & 0.0500 & 0.2420 \end{bmatrix}; \qquad \mathbf{R}_{minimal} = \begin{bmatrix} 1.9229 & 1.8211 & 0.7260 \\ 1.8211 & 1.7400 & 1.1332 \\ 0.7260 & 1.1332 & 6.4104 \end{bmatrix}$$

Since from Figure 5.5 all three controllers seem to perform in a very similar way (especially the minimal case and the controller of Wang and Young (2006) have a

[4]Using the randn MATLAB® function many times, various values of the observer pole positions were obtained. However, the results did not differ leading to the conclusion that such fast observer dynamics do not reflect to the overall closed loop performance.

Table 5.2: Integral of absolute error values for the Shell Heavy Oil Fractionator example for the controller of two parametrisations of the controller described in Chapter 4, the controller of Wang and Young (2006) and a minimal MPC realisation.

Controller	Reference	Cross–Coupling	Total
Chapter 4^a	2.3381	4.2569	6.8896
Chapter 4^b	3.7155	3.1741	6.5950
Wang and Young (2006)	3.1704	6.8514	10.0218
Minimal	3.1650	6.8456	10.0056

visually identical response), Table 5.2 presents the integral–of–error between the actual and desirable system responses. The controller marked with ^{a} refers to the parametrisation presented above. For comparative purposes and to show that even with less optimisation parameters the controller that is based on a system with an explicit integral–of–error state yields improved system responses, the controller marked with ^{b} in Table 5.2 corresponds to parametrisation of the weights as in Section 5.2.2 with:

$$\mathbf{w_y} = diag\begin{bmatrix} 0.2164 & 0.2940 & 1 \end{bmatrix}$$

$$\mathbf{w_{u^-}} = diag\begin{bmatrix} 0.01 & 0.01 & 0.01 \end{bmatrix}$$

$$\mathbf{w_u} = diag\begin{bmatrix} 0.005 & 0.005 & 0.01 \end{bmatrix}$$

and

$$\mathbf{w_z} = \begin{bmatrix} 0.0345 & 0.0122 & 0.0050 \\ 0.0122 & 0.0583 & 0.0050 \\ 0.0050 & 0.0050 & 0.1076 \end{bmatrix}$$

As can be seen from Table 5.2 the alternative controller description of Chapter 4 yields improved results (closer following of the desired response) compared to the controllers of Wang and Young (2006) and the minimal MPC representation for this particular example. Although it may be argued this is due to the

increased number of tuning parameters, the second approach to tuning the the alternative controller of Chapter 4 with less optimisation parameters (marked with ^{b} in Table 5.2) suggests that it is a result of the structure of the controller and not the number of the tuning parameters. The above results and conclusions are summarised in the following section.

5.4 Conclusions

This Chapter proposed a performance optimisation method for MPC controllers based on the goal attainment method. It focused on forming an off–line optimisation problem with optimisation parameters the cost function weightings, that can pose as tuning parameters for the controller. The goal attainment optimisation procedure allows for multiple and even conflicting objectives to be defined that is inherited to the tuning method.

It was shown by simulation examples that the proposed tuning technique can be applied to different MPC structures. It was applied to both SISO and MIMO systems with various design objectives (rise time, settling time, dynamic output decoupling, and specific system dynamics). Moreover, the result of the last simulation suggests that the structure of the alternative controller described in Chapter 4 provides the algorithm with more freedom. This added freedom can be the result of the chosen state vector and the introduction of the integral–of–error state variable.

Concluding, it is claimed that as long as an adequate number of tuning parameters is chosen (and physical constraints allow), the goal attainment method can form an optimal tuning approach in the MPC framework.

Chapter 6

The Forward Path MPC

As has been mentioned in various places in this thesis, recent research on MPC has mainly focused on reducing the on–line computational load resulting from the solution of an optimisation problem of the form (2.4) (e.g. Borrelli et al., 2001; Bartlett et al., 2002; Bemporad et al., 2002; Bacic et al., 2003; Imsland et al., 2005) and to improve it's robustness properties (e.g. Bemporad and Morari, 1999; Kouvaritakis et al., 2000; Wang and Rawlings, 2004; Rodrigues and Odloak, 2005; Mayne et al., 2005).

The use of NMSS models can address these issues as has been presented by Wang and Young (2006). The present chapter extends the results in Wang and Young (2006) by considering two control structures for NMSS/MPC design, namely the conventional 'feedback' form and an alternative 'forward path' structure. The latter approach uses the control model to form part of the state vector, although an outer feedback loop of the measured output variable is still utilised to ensure type one servomechanism performance. The development of both forms is motivated by the analogy with linear PIP control, where each control structure is found to have certain performance and robustness advantages in practice (Taylor et al., 1998).

To investigate the properties of these control structures in the present NMSS context, three simulation examples are considered: a marginally stable, linear

transfer function model; the IFAC'93 benchmark system that has already been presented in Section 4.4.2 (Whidborne et al., 1995); and a multivariable process control example, namely the ALSTOM nonlinear gasifier model, recently utilised as the basis of the IEE Benchmark Challenge II (Dixon and Pike, 2005). Stochastic simulations are utilised to compare the relative robustness of the control structures.

6.1 System and Control Description

The methodological approach reviewed below follows from earlier research into model predictive control (Wang and Young, 2006) based on the representation of the system using NMSS models (Young et al., 1987; Taylor et al., 2000a). It's properties and some control simulations have been extensively studied in Wang and Young (2006), while here it's structure is considered and the control methodology is extended to form the Forward Path NMSS/MPC controller.

Consider again the difference equation (4.1) that allows for embedded integral action in the MPC system, hence ensuring type one servomechanism performance. Defining the state vector as in (4.4), the system can take the following state space form:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\boldsymbol{\Delta}\mathbf{u}_k \tag{6.1a}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k \tag{6.1b}$$

where the $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ matrices are given by (4.6).

The controller is subsequently derived from the numerical minimization of the following index at each sampling instant,

$$J(k, \mathbf{r}_k) = \sum_{i=1}^{N_p} \|\mathbf{r}_{k+i|k} - \mathbf{y}_{k+i|k}\|_{\mathbf{Q}} + \sum_{i=0}^{Nc-1} \|\boldsymbol{\Delta}\mathbf{u}_{k+i|k}\|_{\mathbf{R}} \tag{6.2}$$

subject to the system constraints (4.12b), where $\|\cdot\|_{\mathbf{P}}$ refers to the matrix weighted

squared $2-norm$ of a vector, i.e. $\|\mathbf{x}\|_{\mathbf{P}} = \mathbf{x}^T\mathbf{P}\mathbf{x}$. It should be noted here that (6.2) is equivalent to (4.10), while the former is used here to make the stability analysis that follows more transparent.

Inequalities (4.12b) allow for the definition of system constraints, including both saturation and rate limits on the control input, together with level constraints on the output. Similarly, the cost function (6.2) takes a standard MPC form with a penalty on the future error and input signals. A wide range of numerical solutions to this type of constrained control problem exist (two of which are presented in Sections 2.2.3 and 2.2.4) and have been studied extensively in the literature. In this regard, the examples below utilise the quadprog function from the optimisation toolbox in Matlab®. The receding horizon approach is employed, with only the first element of the predicted input signal $\boldsymbol{\Delta}\mathbf{u}_k$ utilised at each sample k, as presented in Section 2.3.

6.1.1 Stability analysis

The stability properties of MPC are well understood and a general proof has been given in Appendix C. For the approach presented here, the candidate Lyapounov function is expressed in the form (C.2) and reference is made to Theorem C.2.1.

As in Appendix C only stability of the origin is considered while with an appropriate coordinate transformation any equilibrium can be moved to the origin. In this regard, the following assumptions are made.

Theorem 6.1.1. *Under the Assumptions C.2.1 and C.2.2 the Model Predictive Control scheme presented here is asymptotically stable.*

Proof. The minimum of the finite horizon cost function (6.2) is chosen as a candi-

date Lyapounov function:

$$\begin{aligned} V(\mathbf{x}_k, \boldsymbol{\Delta}\mathbf{u}^*_{|k\rightarrow}) &= \sum_{i=1}^{N_p} \|\mathbf{r}_{k+i|k} - \mathbf{y}_{k+i|k}\|_{\mathbf{Q}} + \sum_{i=0}^{Nc-1} \|\boldsymbol{\Delta}\mathbf{u}_{k+i|k}\|_{\mathbf{R}} \\ &= \sum_{i=1}^{N_p} \|\mathbf{C}\mathbf{x}_{k+i|k}\|_{\mathbf{Q}} + \sum_{i=0}^{Nc-1} \|\boldsymbol{\Delta}\mathbf{u}_{k+i|k}\|_{\mathbf{R}} \\ &= \sum_{i=1}^{N_p} \mathbf{x}^T_{k+i|k}\mathbf{C}^T\mathbf{Q}\mathbf{C}\mathbf{x}_{k+i|k} + \sum_{i=0}^{N_c-1} \boldsymbol{\Delta}\mathbf{u}^{*T}_{k+i|k}\mathbf{R}\boldsymbol{\Delta}\mathbf{u}^*_{k+i|k} \end{aligned}$$

It should be noted that $\mathbf{r}_{k+i|k} = \mathbf{0}$ since the origin is considered the operating point. The above is in the form (C.2) with the decision variables being $\boldsymbol{\Delta}\mathbf{u}^*_{|k\rightarrow}$ and the weighting matrix of the predicted states being $\mathbf{C}^T\mathbf{Q}\mathbf{C}$. Therefore, from Theorem C.2.1 it follows that the system is asymptotically stable, which completes the proof. □

6.1.2 Feasibility

Many authors have addressed the problem of feasibility and several potential solutions to this problem are briefly discussed here. An obvious approach is to soften the output constraints by dropping them and instead introducing a constraint violation penalty in the cost function (see Section 3.4 of Maciejowski, 2002, for an example). This method of regaining feasibility using slack variables, together with a way to guarantee that the constraints will be satisfied if possible, is also discussed by Kerrigan and Maciejowski (2000). Another approach, considered by Rawlings and Muske (1993), identifies the smallest time beyond which the constraints can be satisfied and enforces the constraint after that time. By contrast, Scokaert and Rawlings (1999) regard constraint violation as a multi–objective problem.

In the cases presented in this thesis, both input and input rate–of–change constraints are considered saturating and therefore cannot be violated. However, in cases of large disturbances or uncertainty the output constraints may be violated (they are considered as constraints posed by performance or safety issues). To

deal with such cases any output constraints are simply removed at the time of infeasibility so that the control structure does not become complicated and the effect of the structure is more apparent (and not the effect of the infeasibility handling techniques). When the system is returned to the region where the quadratic program becomes feasible, the constraints are reinstated.

6.2 Control structures

In order to develop new control structures for implementing the above algorithm, consider in the first instance the *unconstrained* solution. By analytically calculating the derivative of (6.2), the incremental control signal at time k becomes,

$$\mathbf{\Delta u}_k = -\mathbf{K}_s\mathbf{x}_k + \mathbf{r}_s \tag{6.3}$$

in which $\mathbf{r}_s$ is a $q \times 1$ column vector accounting for set-point changes and the state feedback gain matrix takes the following block matrix form,

$$\mathbf{K}_s = \begin{bmatrix} \mathbf{K}_1 & \mathbf{K}_2 & \cdots & \mathbf{K}_n & \mathbf{K}_{n+1} & \cdots & \mathbf{K}_{m+n-1} & \mathbf{K}_{m+n} \end{bmatrix}$$

where,

$$\mathbf{K}_i = \begin{cases} \begin{bmatrix} k_{i,1}^1 & k_{i,2}^1 & \cdots & k_{i,p}^1 \\ k_{i,1}^2 & k_{i,2}^2 & \cdots & k_{i,p}^2 \\ \vdots & \vdots & \ddots & \vdots \\ k_{i,1}^q & k_{i,2}^q & \cdots & k_{i,p}^q \end{bmatrix} & i = 1, \ldots, n \text{ or } i = m+n \\ \begin{bmatrix} k_{i,1}^1 & k_{i,2}^1 & \cdots & k_{i,q}^1 \\ k_{i,1}^2 & k_{i,2}^2 & \cdots & k_{i,q}^2 \\ \vdots & \vdots & \ddots & \vdots \\ k_{i,1}^q & k_{i,2}^q & \cdots & k_{i,q}^q \end{bmatrix} & i = n+1, \ldots, m+n-1 \end{cases}$$

in which $\mathbf{K}_1 \cdots \mathbf{K}_{m+n}$ are the control gain submatrices. Equation (6.3) is particularly straightforward to implement in the NMSS case, since $\mathbf{x}_k$ is directly available for feedback and hence an observer is not required.

6.2.1 Feedback structure

At this juncture, it is useful to represent the MPC algorithm in polynomial form, so that equivalent block diagrams can be developed. In this regard, consider the following matrices with polynomial functions as elements,

$$\begin{aligned} \mathbf{P}(z) &= \begin{bmatrix} \mathbf{P}_1(z) & \mathbf{P}_2(z) & \cdots \mathbf{P}_p(z) \end{bmatrix} \\ \mathbf{L}(z) &= \begin{bmatrix} \mathbf{L}_1(z) & \mathbf{L}_2(z) & \cdots \mathbf{L}_q(z) \end{bmatrix} \end{aligned}$$

where,

$$\mathbf{P}_i(z) = \begin{bmatrix} k^1_{1,i} + k^1_{2,i}z^{-1} + k^1_{3,i}z^{-2} + \cdots + k^1_{n,i}z^{-(n-1)} \\ k^2_{1,i} + k^2_{2,i}z^{-1} + k^2_{3,i}z^{-2} + \cdots + k^2_{n,i}z^{-(n-1)} \\ \vdots \\ k^q_{1,i} + k^q_{2,i}z^{-1} + k^q_{3,i}z^{-2} + \cdots + k^q_{n,i}z^{-(n-1)} \end{bmatrix}$$

and

$$\mathbf{L}_i(z) = \begin{bmatrix} 1 + k^1_{n+1,i}z^{-1} + k^1_{n+2,i}z^{-2} + \cdots + k^1_{n+m-1,i}z^{-(m-1)} \\ 1 + k^2_{n+1,i}z^{-1} + k^2_{n+2,i}z^{-2} + \cdots + k^2_{n+m-1,i}z^{-(m-1)} \\ \vdots \\ 1 + k^q_{n+1,i}z^{-1} + k^q_{n+2,i}z^{-2} + \cdots + k^q_{n+m-1,i}z^{-(m-1)} \end{bmatrix}$$

The feedback control law is subsequently written as follows,

$$\left(1 - z^{-1}\right)\mathbf{L}(z)\mathbf{U}(z) = \mathbf{w}_s\mathbf{R}(z) - \left(1 - z^{-1}\right)\mathbf{P}(z)\mathbf{Y}(z) - \mathbf{K}_{n+m}\mathbf{Y}(z)$$

where $\mathbf{w}_s\mathbf{r}_k = \mathbf{r}_s$ and the command vector $\mathbf{r}_k$ is the inverse z-transform of $\mathbf{R}(z)$. It is straightforward to show that $\mathbf{w}_s = \mathbf{K}_{m+n}$ and, therefore, the control law takes

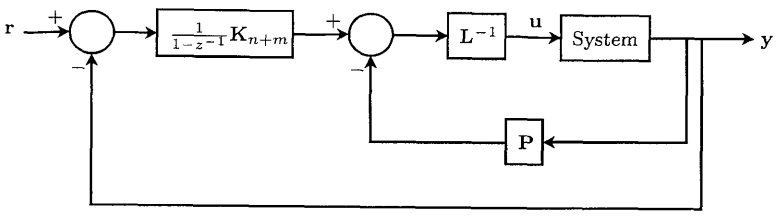

Figure 6.1: NMSS/MPC represented in feedback form.

the form,

$$\mathbf{U}(z) = \mathbf{L}^{-1}(z)\left(\frac{1}{(1-z^{-1})}\mathbf{K}_{n+m}\left(\mathbf{R}(z) - \mathbf{Y}(z)\right) - \mathbf{P}(z)\mathbf{Y}(z)\right) \tag{6.4}$$

The above NMSS/MPC control strategy is illustrated in Figure 6.1 and in terms of state variable feedback is clearly in a feedback form.

6.2.2 Forward path form

Equation (6.4) exposes the matrix of integral gains for each input–output pathway $\mathbf{K}_{n+m}$, i.e. that part of the control gain matrix associated with the outputs $\mathbf{y}$ in the NMSS state vector. As a consequence, in order to achieve type one servomechanism performance, the last p elements of the state vector should always be the directly measured values of the outputs.

However, this is not strictly necessary for the other elements in the state vector. In fact, following a similar approach to PIP control system design (Taylor et al., 1998), the other state variables can be replaced with those obtained from the control model. The forward path formulation that this approach yields is illustrated in Figure 6.2, where the left Matrix Fraction Description (MFD) (Kailath, 1980) of the control model is defined by $\hat{\mathbf{S}}(z)$, i.e. $\mathbf{y}(k) = \hat{\mathbf{S}}(z)\mathbf{u}(k)$. In this case, the closed loop MFD relating the command input to the output variable, $\mathbf{y}_k = \mathbf{T}(z)\mathbf{r}_k$,

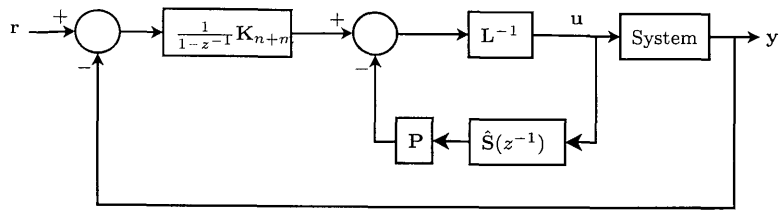

Figure 6.2: NMSS/MPC represented in forward path form.

is defined as follows,

$$\mathbf{T}(z) = \left[(1-z^{-1})\mathbf{I}_p + \mathbf{S}(z)\left(\mathbf{I}_q + \mathbf{L}^{-1}(z)\mathbf{P}(z)\hat{\mathbf{S}}(z)\right)^{-1}\mathbf{L}^{-1}(z)\mathbf{K}_{n+m}\right]^{-1} \left[\mathbf{S}(z)\left(\mathbf{I}_q + \mathbf{L}^{-1}(z)\mathbf{P}(z)\hat{\mathbf{S}}(z)\right)^{-1}\mathbf{L}^{-1}(z)\mathbf{K}_{n+m}\right] \tag{6.5}$$

It follows that $\lim_{z\to 1}\mathbf{T}(z) = \mathbf{I}_{p\times p}$, hence the closed loop system tracks a constant set point change without steady state error. Similarly, the closed loop MFD for an input disturbance to the output, $\mathbf{y}_k = \mathbf{S}_i(z)\mathbf{d}_{u,k}$, is,

$$\mathbf{S}_i(z) = \left[(1-z^{-1})\mathbf{I}_p + \mathbf{S}(z)\left(\mathbf{I}_q + \mathbf{L}^{-1}(z)\mathbf{P}(z)\hat{\mathbf{S}}(z)\right)^{-1}\mathbf{L}^{-1}(z)\mathbf{K}_{n+m}\right]^{-1} \left[(1-z^{-1})\mathbf{S}(z)\left(\mathbf{I}_q + \mathbf{L}^{-1}(z)\mathbf{P}(z)\hat{\mathbf{S}}(z)\right)^{-1}\right] \tag{6.6}$$

while the equivalent MFD for an output disturbance, $\mathbf{y}_k = \mathbf{S}_o(z)\mathbf{d}_{y,k}$, is,

$$\mathbf{S}_o(z) = \left[(1-z^{-1})\mathbf{I}_p + \mathbf{S}(z)\left(\mathbf{I}_q + \mathbf{L}^{-1}(z)\mathbf{P}(z)\hat{\mathbf{S}}(z)\right)^{-1}\mathbf{L}^{-1}(z)\mathbf{K}_{n+m}\right]^{-1} \left[(1-z^{-1})\mathbf{I}_p\right] \tag{6.7}$$

From equations (6.6) and (6.7) it follows that,

$$\begin{aligned} \lim_{z\to 1}\mathbf{S}_i(z) &= \mathbf{0}_{p\times q} \\ \lim_{z\to 1}\mathbf{S}_o(z) &= \mathbf{0}_{p\times p} \end{aligned}$$

Therefore, the closed loop system will reject any constant disturbance.

6.2.3 Constraints

In the presence of constraints, the control action is still derived by numerical optimisation of (6.2) subject to constraints (4.12b). However, in the forward path case, the $\Delta\mathbf{x}_{m,k}$ part of the current measurement of the state vector (4.4) is replaced by $\Delta\hat{\mathbf{x}}_{m,k}$, formed from the output of the estimated system $\hat{\mathbf{S}}(z^{-1})$ and not the actual outputs of the system (i.e. $\hat{\mathbf{y}}_k = \hat{\mathbf{S}}(z)\mathbf{u}_k$) as follows:

$$\Delta\hat{\mathbf{x}}_{m,k}^T = \left[\Delta\hat{\mathbf{y}}_k^T \quad \Delta\hat{\mathbf{y}}_{k-1}^T \quad \cdots \quad \Delta\hat{\mathbf{y}}_{k-n+1}^T \quad \Delta\mathbf{u}_{k-1}^T \quad \Delta\mathbf{u}_{k-2}^T \quad \cdots \quad \Delta\mathbf{u}_{k-m+1}^T\right]$$

The quadprog function of the Matlab® identification toolbox is subsequently evaluated to solve the resulting quadratic program, as mentioned above.

Although the set–point following and disturbance rejection results have been developed for the unconstrained case, the following remark now considers the general solution when constraints are present. The remark assumes only that the reference levels have been realistically chosen, i.e. they can be reached in practice given the input constraints.

Remark 6.2.1. *When the outputs of the system are near the reference levels, the controller has the unconstrained form. Furthermore, the set-point following and disturbance rejection properties presented above, all refer to the steady state. Therefore, they also hold in the general case where constraints are present.*

6.3 Performance tests

In this section, the performance of the proposed forward path NMSS/MPC algorithm is compared with that of the feedback form, as applied to three simulation examples. The research takes advantage of Monte Carlo Simulation (MCS), which provides one of the simplest and most attractive approaches to assessing the sensitivity of a controller to parametric uncertainty. Here, the model parameters for

each stochastic realisation are selected randomly from a given probability distribution.

6.3.1 Monte Carlo Simulation (MCS) based uncertainty analysis

It can be shown (Young, 1976, 1984) that the RIV and SRIV estimation algorithms introduced in Section 2.1 are optimal in statistical terms if the additive noise has a normal Gaussian distribution. In this situation, the parameter estimates will have a joint probability distribution with mean and covariance matrix defined by the estimated $r \times 1$ parameter vector $\hat{\mathbf{p}}$ and its associated $r \times r$ covariance $\mathbf{P}^*$. In MCS analysis (e.g. Taylor et al., 2004b; Young, 1999), these statistical results are used to generate a large number or 'ensemble' N, of stochastically generated random samples, or 'realisations', of the model parameters from the multivariate normal distribution defined by $\hat{\mathbf{p}}$ and $\mathbf{P}^*$.

In order to compute the random ensemble of model parameters that can be considered to characterise the model in a stochastic sense, it is necessary to generate random numbers from the normal distribution $\mathcal{N}(\hat{\mathbf{p}}, \mathbf{P}^*)$. This is accomplished by noting that $\mathbf{P}^*$ is a positive definite matrix, so that it can be decomposed using the Cholesky decomposition,

$$\mathbf{P}^* = \mathbf{L}\mathbf{L}^T$$

where the Cholesky factor $\mathbf{L}$ is a nonsingular $r \times r$ lower triangular matrix. Consider now the set of N random vectors,

$$\tilde{\mathbf{p}}_i = \mathbf{L}\mathbf{e}_i \qquad , i = 1, \cdots, N \tag{6.8}$$

where $\mathbf{e}_i = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is an $r \times 1$ vector of independent, normally distributed random variables, each with zero mean and unity variance. By construction, the vectors

$\tilde{\mathbf{p}}_i$, $i = 1, \cdots, N$ have the properties:

$$E(\tilde{\mathbf{p}}_i) = 0$$

$$\text{cov}(\tilde{\mathbf{p}}_i) = E(\tilde{\mathbf{p}}_i \tilde{\mathbf{p}}_i^T) = \mathbf{L} E(\mathbf{e}_i \mathbf{e}_i^T) \mathbf{L}^T = \mathbf{L}\mathbf{L}^T$$

The vector $\tilde{\mathbf{p}}_i$, generated in this manner, has the same covariance matrix as the parameter estimates. Consequently, the vector,

$$\mathbf{p}_i^* = \hat{\mathbf{p}} + \tilde{\mathbf{p}}_i \tag{6.9}$$

can be considered as a random sample from the estimated parametric probability distribution $\mathcal{N}(\hat{\mathbf{p}}, \mathbf{P}^*)$, as required for MCS analysis.

In the present context, the reduced order model is obtained by performing a planned, open loop experiment on the large system model using selected perturbational input signals. Since the noise level is low in such an experiment, the SRIV algorithm is used both to identify the reduced order model structure that describes the data adequately and estimate its parameters, together with their associated covariance matrix. Given this information, MCS analysis can be used to evaluate how the uncertainty in the parameter estimates affects other properties of the open or closed loop system: for instance, plots of the closed loop step, impulse and frequency responses; and plots of the closed loop pole positions or 'stochastic root loci' (e.g. Taylor et al., 2004b).

The computational implementation of MCS analysis is straightforward. For the i^{th} realisation, the r elements of the white noise vector $\mathbf{e}_i$ are generated as independent, normally distributed random numbers using, for example, the randn function in Matlab®. The Cholesky transformation in (6.8) is then applied to obtain $\tilde{\mathbf{p}}_i$, while the parameter vector $\mathbf{p}_i^*$ from (6.9) defines the randomly selected model. This model is then used to compute the i^{th} realisation of the model property being investigated. After this has been repeated N times, the resulting ensemble of

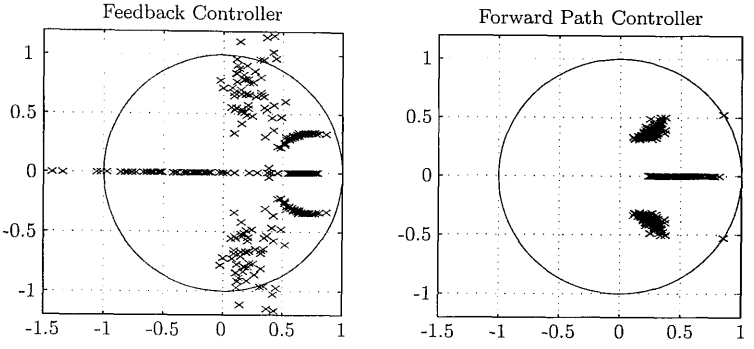

Figure 6.3: Root loci for 100 Monte Carlo simulations for 15% variation in each numerator parameter of the transfer function (6.10). Left figure: feedback form. Right figure: forward path form.

N realisations can be plotted directly, as shown in the examples considered later, or they can be represented by their statistical properties (e.g. Young, 1999): these could include an empirical estimate of the probability distribution for a derived parameter in the form of a normalised histogram; the mean and 95 percentile bounds of a response; (etc.). Naturally, the accuracy of these empirical estimates is a function of the ensemble size N but, in these days of fast digital computers, it is normally possible to choose this to be sufficiently large.

6.3.2 Marginally stable system

Consider the second order, marginally stable and non–minimum phase model introduced in Section 3.4 in the following transfer function form,

$$S(z^{-1}) = \frac{-z^{-1} + 2z^{-2}}{1 - 1.7z^{-1} + z^{-2}} \tag{6.10}$$

Since the above system is marginally stable, introducing uncertainty is likely to cause instability of the closed loop system. To evaluate the performance of the structures presented above, they are both applied on this system. For the purposes of this example, the prediction and control horizons are selected as $N_p = 20$ and $N_c = 5$. The $\mathbf{Q}$ and $\mathbf{R}$ matrices are $\frac{1}{20}$ and $\frac{1}{5}$, respectively, chosen by trial and error to obtain a satisfactory response for the nominal case.

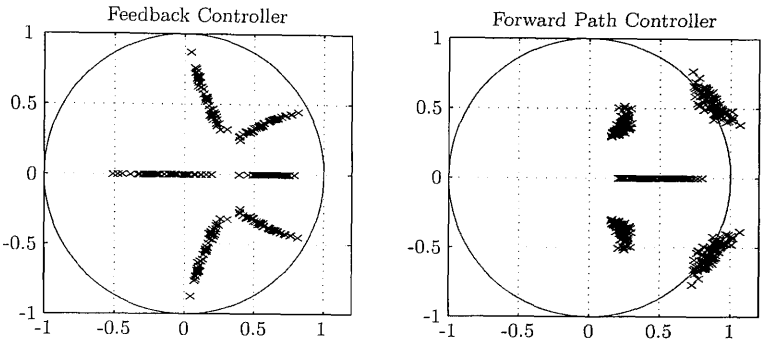

Figure 6.4: Root loci for 100 Monte Carlo simulations for 5% variation in each denominator parameter of the transfer function (6.10). Left figure: feedback form. Right figure: forward path form.

With these settings and 100 MCS realisations performed in both MPC structures presented in Section 6.2, Figure 6.3 shows the closed loop transfer function poles for 15% variation in each numerator parameter (i.e. every numerator parameter was randomly varied within $\pm 15\%$ of it's nominal value), while Figure 6.4 illustrates the case for 5% variation in the denominator parameters. For these initial simulations, the level of parametric uncertainty has been artificially chosen, in order to highlight the difference between the two control structures and to illustrate the way the closed loop poles are affected by parameter variation.

In both this and the following example, such closed loop pole positions refer to the poles of (6.5). Although these are for the unconstrained case, they provide insight into the behaviour of both controller structures under uncertainty. The NMSS/MPC pole positions presented in Figures 6.3 and 6.4 mirror results obtained from earlier research into NMSS/PIP control (Taylor et al., 1998, 2001), i.e. the forward path controller yields more unstable responses in the case of parametric uncertainty of the denominator, while the feedback structure yields more unstable responses when there is only uncertainty in the numerator parameters.

By contrast, Figure 6.5 illustrates 100 MCS realisations for simultaneous variations in both the numerator and denominator parameters, based on the estimated covariance matrix, as discussed in Section 6.3.1 above. In this case, the analysis utilises $\mathbf{P}^*$ identified from an initial open loop experiment with artificially intro-

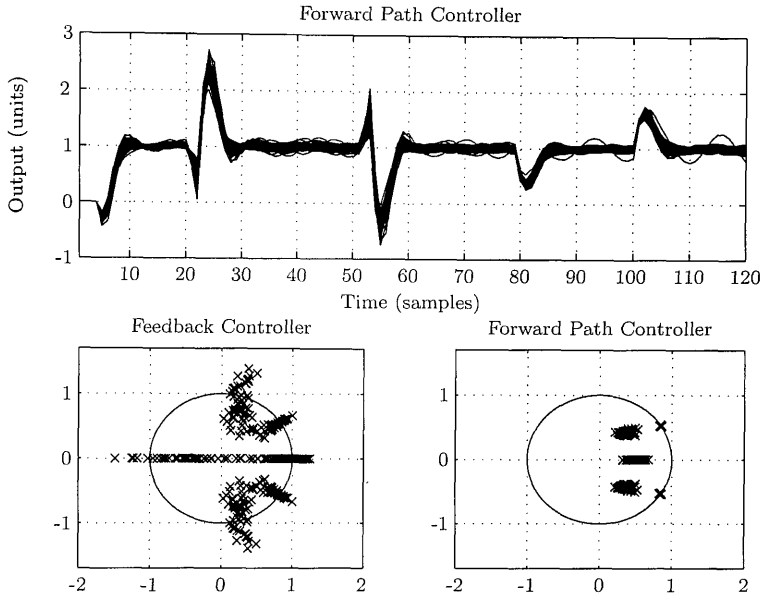

Figure 6.5: 100 Monte Carlo simulations for variation in both the numerator and denominator parameters of the transfer function (6.10) based on the estimated $\mathbf{P}^*$ matrix. Top figure: forward path form, also showing the response to input and output disturbances. Bottom left figure: root loci for the feedback form. Bottom right figure: root loci for the forward path form.

duced, additive white noise and is given by:

$$\mathbf{P}^* = \begin{bmatrix} 0.4 & -0.33 & -4.2 & 7.9 \\ -0.33 & 0.38 & -1.2 & -2.9 \\ -4.2 & -1.2 & 410 & -380 \\ 7.9 & -2.9 & -380 & 440 \end{bmatrix} \cdot 10^{-4}$$

The feedback controller yields numerous highly unstable realisations, as illustrated by the closed loop pole positions. For this reason, the upper subplot of Figure 6.5 only shows the time responses of the forward path controller. Figure 6.5 represents the most realistic comparison of the two controllers, from which it is clear that the forward path structure yields the most robust performance. However, it is important to note that, not only has a difficult marginally stable system (6.10) to control been chosen, but the parametric uncertainty has been artificially scaled up, in order to facilitate this comparison of the relative robustness.

Figure 6.5 also illustrates the disturbance rejection properties of the proposed forward path algorithm. Here, a pulse input disturbance of length 30 samples and amplitude 0.2 is initiated at the 20th sample, while a pulse output disturbance of 20 samples length and amplitude -0.5 starts at the 80th sample. As would be expected from the analysis in Section 6.2.2, the algorithm satisfactorily rejects both these disturbances in steady state.

Finally, for all the examples above, the forward path structure generally yields closed loop poles in the vicinity of the theoretical design poles, while in the case of the feedback controller the poles are more widely distributed. When the design poles are well inside the unit circle, this might be expected to yield stable MCS forward path responses closer to the design specifications, as shown by the next example.

6.3.3 The IFAC '93 Benchmark

As already mentioned in Section 4.4.2, the IFAC '93 benchmark is a difficult problem since it includes stringent closed loop performance requirements, despite the fact that the parameters of the plant are known only within a certain range. In fact, the plant operates at three different stress levels, with higher stress levels implying greater time variations of the unknown parameters and a continuous time transfer function representation takes the form of (4.22).

For brevity, the present section does not repeat the system description that has already been presented in Section 4.4.2, the reduced order transfer function for the model is given by (4.23). For the control simulations presented in the following, the prediction and control horizons, together with the weighting terms are chosen as follows: $N_p = 50$, $N_c = 10$, $\mathbf{Q} = \frac{10}{50}$ and $\mathbf{R} = \frac{1}{10}$.

The benchmark transfer function (4.22) is already a reduced order model (see Whidborne et al., 1995, for details), hence the $\mathbf{P}^*$ matrix (see Section 6.3.1) associated with (4.23) is not necessarily an appropriate representation of the true

Table 6.1: Lower and upper bounds of parameter uncertainty that result in the stochastic closed loop poles lying within 0.1 of the nominal case for the IFAC '93 Benchmark, where ϵ_1 and ϵ_2 are given by equation (6.11).

	ϵ_1		ϵ_2	
	Min.	Max.	Min.	Max.
Feedback	0.837	1.126	0.939	1.105
Forward Path	0.735	1.449	0.868	1.071

uncertainty. However, for the purposes of the present chapter, it is the relative performance of the two control structures that is of most interest; for this example, multiplicative uncertainty is introduced into the numerator and denominator parameters of (4.23) in the following manner:

$$\begin{aligned} b_2 &= \epsilon_1 \hat{b}_2 \\ a_1 &= \epsilon_2 \hat{a}_1 \end{aligned} \tag{6.11}$$

in which $\hat{b}_2$ and $\hat{a}_1$ are the values that were used while designing the controllers (i.e. $\hat{b}_2 = 0.0946$ and $\hat{a}_1 = -0.9055$). The lower and upper values of ϵ_1 and ϵ_2 that allow for all the poles of the uncertain closed loop to lie inside a circle of radius 0.1, with the centre defined as the associated nominal closed loop pole, are determined numerically and shown in Table 6.1. This problem is solved using the fmincon function of the MATLAB® optimisation toolbox, defining the polynomial roots as nonlinear constraints.

For this system, Table 6.1 shows that the forward path controller yields closed loop poles 'near' the nominal case for a wider range of the uncertain parameters than the feedback structure. For example, the above condition holds for b_2 in the range 0.74 to 1.45 of the nominal value for the forward path structure, compared with a more constrained range of 0.84 to 1.13 in the feedback case. It should be noted that only one of ϵ_1 and ϵ_2 is allowed to vary at a time, since simultaneous variation of both parameters does not allow the formulation of an optimisation problem that can be solved without compromise between variation of the parame-

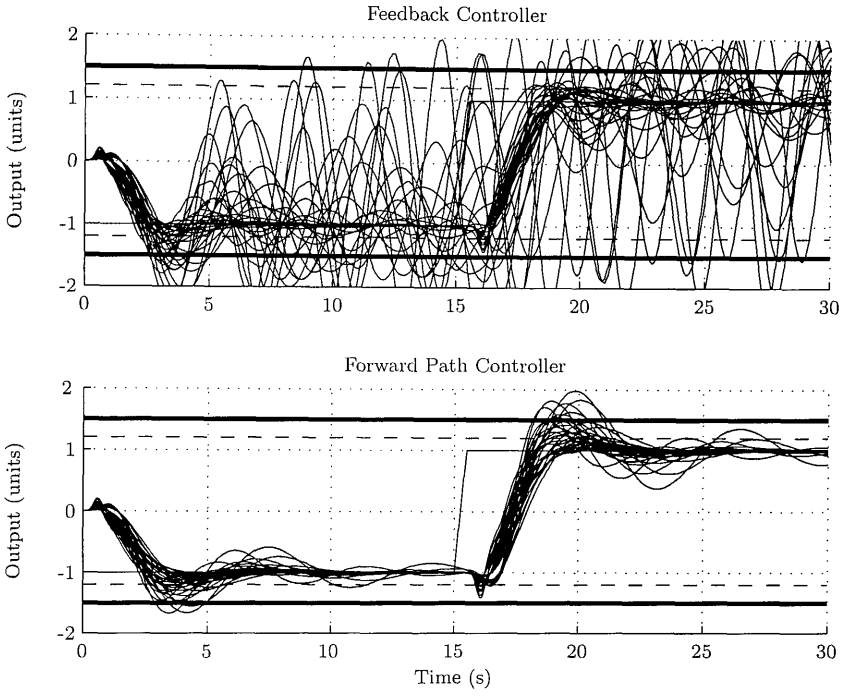

Figure 6.6: Closed–loop simulations of the IFAC '93 benchmark with the parametric variation at Stress Level 2 as shown in Table 4.1. Top figure: feedback form. Bottom figure: forward path form.

ters. By contrast, the additional simulations described in the following paragraph apply to variations of the parameters of system (4.22).

To support the results above, 32 simulations were obtained based on the positive and negative extremes of each of the 5 stochastic parameters (see Table 4.1), now using the benchmark system (4.22). Although it is possible that more extreme responses may be obtained with certain combinations of the parameters in their intermediate range, the simulations employed here provide a reasonable guide to the overall performance. The results are illustrated in Figure 6.6, where it is clear that the forward path controller yields an improved response (with no unstable realisations) compared to the feedback algorithm at the specified level of parametric uncertainty.

It should be noted that the objective of the present simulations are only to use Figure 6.6 to highlight the difference between the two control structures and not to present a finally designed control system. For instance, some of the realisations,

even in the case of the forward path algorithm, have relatively large excursions from the command level.

In this regard, the performance shown in Figure 6.6 for the forward path controller is similar to the initial fixed gain NMSS/PIP controller developed in Taylor et al. (2001). In this latter case, a multi–objective optimisation approach was subsequently utilised to significantly improve the results (Taylor et al., 2001). This approach has been presented for MPC controllers in Chapter 5 and clearly offers a solution to tuning the present controller. To further improve the robustness of the Forward Path controller, the decrease of the overshoot of the worst case in the robustness test is identified as a performance requirement, while at the same time the rise time of the slowest response should not be dramatically deceased. The optimisation parameters are chosen as the evolution of the $\mathbf{Q}$ and $\mathbf{R}$ matrices (i.e. elements in the present SISO example) within the prediction horizon. Defining the parametrisation of (5.4) for the $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ matrices, the optimised values are the following:

$$\bar{\mathbf{Q}} = diag\,[0.4394 \quad 0.1545 \quad 0.1465 \quad 0.2611 \quad 0.4041 \quad 0.4914 \quad 0.5561 \quad 0.6027 \quad 0.6354 \quad 0.6572]$$

and

$$\bar{\mathbf{R}} = diag\,\begin{bmatrix} 0.001 & 0.9284 & 0.9072 & 0.8042 & 0.6891 \end{bmatrix}$$

Finally, the resulting optimised closed loop robustness test, along with the respective control signals is depicted in Figure 6.7, where it is clear that all the design objectives are satisfied.

6.3.4 The ALSTOM Benchmark

This section of the chapter applies the NMSS/MPC approach to the nonlinear ALSTOM Benchmark Challenge II (Dixon and Pike, 2005). The simulation includes

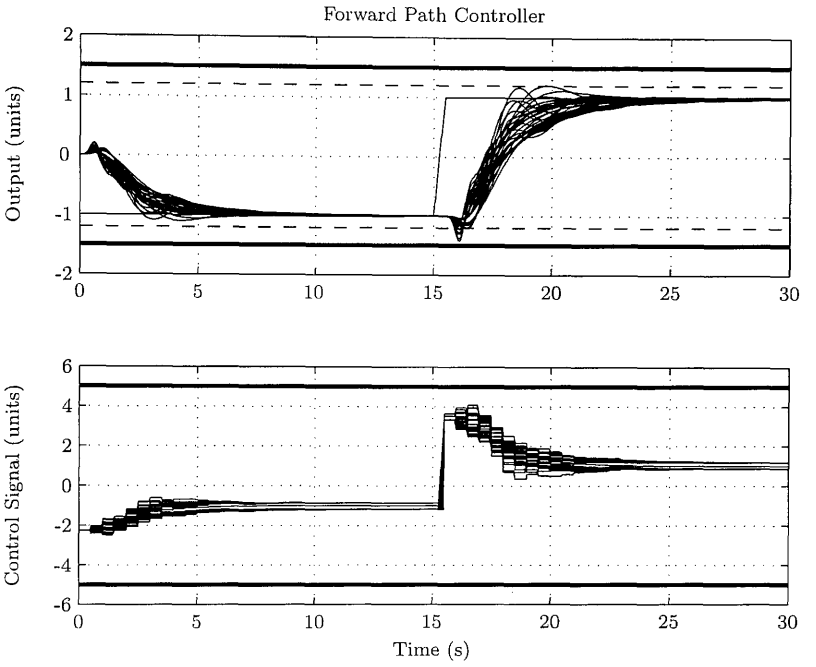

Figure 6.7: Closed loop simulations and control signals for the IFAC '93 benchmark with the parametric variation at Stress Level 2 as shown in Table 4.1 and multi–objective optimisation of the weighting parameters of the Forward Path Controller. Top figure: System Output. Bottom figure: Control signal.

all the significant physical effects related to the gasification system of an integrated gasification combined cycle power plant; e.g. drying processes, desulphurization and pyrolysis. It has previously been validated against measured time histories from a British Coal experimental test facility.

In essence, the benchmark is a nonlinear, multivariable simulation based on four highly constrained controllable inputs, i.e. a linked coal and limestone variable (WCOL), air (WAIR), steam (WSTM) and char extraction (WCHR), together with a pressure disturbance and various boundary conditions, including a coal quality parameter. The pressure (PGAS), temperature (TGAS), bed-mass (MASS) and gas quality (CVGAS) must all be maintained within specified limits, despite the effects of the disturbance signal.

To identify an appropriate control model, the gasifier simulation is first per-

turbed by impulse input signals in open loop. A sampling rate of 1s is utilised[1] to make the results directly comparable with those of Al Seyab and Cao (2006), where it is suggested that the present controller, although linear, can result in improved performance when compared to the nonlinear one of Al Seyab and Cao (2006). In this case, the SRIV algorithm, coupled with the YIC and R_T^2 identification criteria (see Section 2.1), suggest that the gasifier is well represented by four multi-input, single-output transfer function models. In this manner, the following system description is developed,

$$\Delta \mathbf{y}_k + \mathbf{A}_1 \Delta \mathbf{y}_{k-1} + \mathbf{A}_2 \Delta \mathbf{y}_{k-2} + \mathbf{A}_3 \Delta \mathbf{y}_{k-3} = \mathbf{B}_1 \Delta \mathbf{u}_{k-1} + \mathbf{B}_2 \Delta \mathbf{u}_{k-2} \qquad (6.12)$$

where $\Delta \mathbf{y}_k$ consists of the PGAS, TGAS, MASS and CVGAS variables; $\Delta \mathbf{u}_k$ is a vector of inputs, WCOL, WAIR, WSTM and WCHR; and,

$$\mathbf{A}_1 = \text{diag}\begin{bmatrix} -1.8225 & -1.7652 & -0.8469 & -0.9984 \end{bmatrix}$$

$$\mathbf{A}_2 = \text{diag}\begin{bmatrix} 0.8298 & 0.7663 & 0.1195 & 0 \end{bmatrix}$$

$$\mathbf{A}_3 = \text{diag}\begin{bmatrix} 0 & 0 & -0.0268 & 0 \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} 0.0135 & -8.3538 & 3.0317 & -5.7507 \\ -0.0773 & -0.1199 & 0.0734 & -0.1603 \\ 0.0001 & 0.0271 & 0.0066 & 0.0434 \\ 0.0174 & 0.0405 & -0.0371 & -0.0139 \end{bmatrix}$$

$$\mathbf{B}_2 = \begin{bmatrix} 0 & 7.4359 & -2.4752 & 7.0412 \\ 0 & 0 & 0 & 0 \\ 0 & -0.0065 & 0 & -0.0079 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[1] A shorter sampling time of 0.25s seems to be a better choice and to offer an adequate description of the short term dynamics and ensures a rapid response to the disturbances as described in Taylor and Shaban (2006). Although the results presented here are within the specified limits, using faster sampling leads to improvement in the controller performance. However, a sampling time of 1s is chosen for the reasons stated in the main body of the text.

Table 6.2: Boundary values for the control signals of the ALSTOM nonlinear gasifier system.

Input	Amplitude		Rate-of-change	
	Min.	Max.	Min.	Max.
u_1	0	3.5	-0.2	0.2
u_2	0	20	-1	1
u_3	0	10	-0.2	0.2
u_4	0	6	-1	1

The prediction and control horizons are selected as $N_p = 20$ and $N_c = 7$ respectively; the weighting matrices on the error and the control increment signals are given by $\mathbf{Q} = \text{diag}\,[0.25 \quad 0.1 \quad 1000 \quad 10]$ and $\mathbf{R} = \text{diag}\,[1 \quad 10 \quad 0.1 \quad 10]$ respectively. Finally, the level and rate-of-change constraints on the control signals are listed in Table 6.2 (Dixon and Pike, 2005).

Figures 6.8 and 6.9 illustrate the output and input signals, respectively, for the forward path controller, showing the response of the closed-loop system to a sine wave pressure disturbance. This simulation experiment is one of the standard benchmark tests (Dixon and Pike, 2005). It is clear that all four output variables are controlled well within the required limits, where the latter are shown as the horizontal dotted traces in Figure 6.8. Note that the basic performance specifications are met for both control structures although, for brevity, the response of the feedback algorithm is not plotted here.

As expected, the control inputs also remain within the specified level limits at all times, similarly represented as dotted traces in Figure 6.9. However, it is the *rate-of-change* constraints on the control signals for both the feedback and the forward path forms of the MPC algorithm that are of most interest to the present example, illustrated in Figures 6.10 and 6.11 respectively. Here, it is clear that the rate–of–change for the inputs of the forward path controller are considerably smaller than in the feedback case, representing a rather smoother control signal. For a practical system, the latter result suggests that the forward path design may yield reduced wear of the actuators, with the potential for lower operating costs

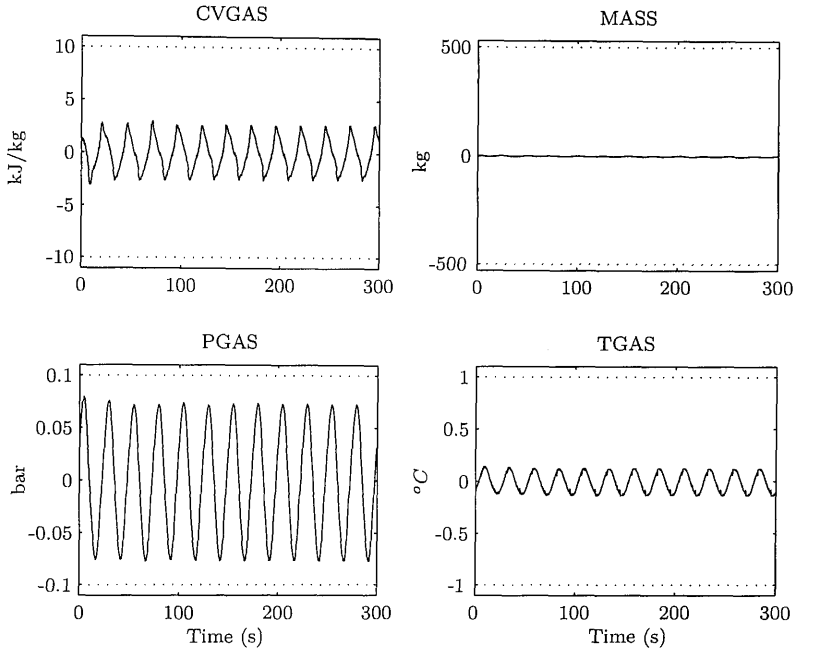

Figure 6.8: System outputs of the ALSTOM gasifier in response to a pressure sine wave disturbance at 100% load using the forward path form.

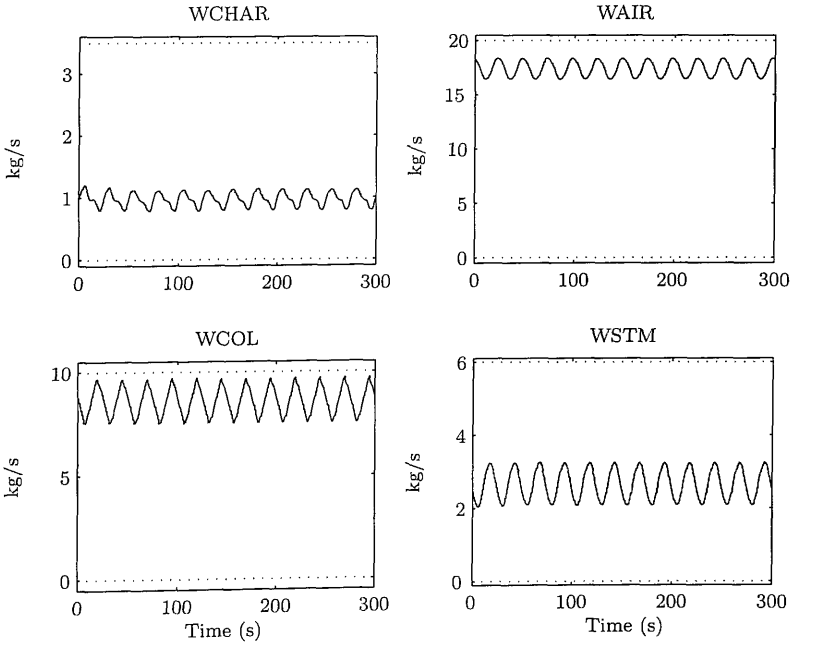

Figure 6.9: System inputs of the ALSTOM gasifier in response to a pressure sine wave disturbance at 100% load using the forward path form.

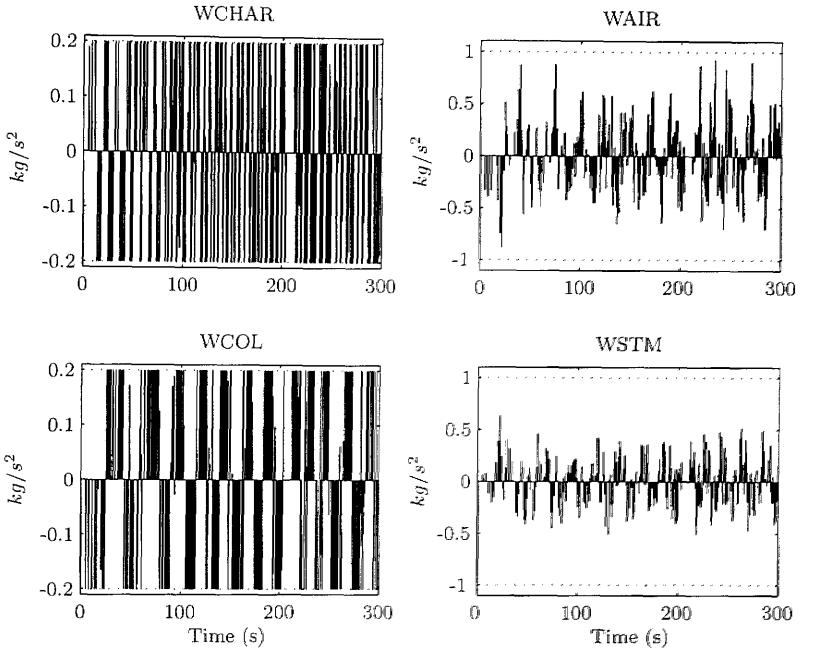

Figure 6.10: Rate–of–change for the four control input signals of the ALSTOM gasifier system at 100% load using the feedback form.

Table 6.3: Integral of absolute error values for the ALSTOM nonlinear gasifier system for the NMSS/MPC controller and the controller of Al Seyab and Cao (2006) (NMPC), shown for three operating levels.

Output	0% load		50% load		100% load	
	NMSS	NMPC	NMSS	NMPC	NMSS	NMPC
CVGAS	705.31	295.29	316.51	760.38	398.95	898.07
MASS	430.63	9292.00	287.20	1010.70	211.91	434.99
PGAS	25.25	13.61	17.61	6.41	13.65	5.24
TGAS	51.81	101.38	32.13	66.17	22.85	45.95

at no loss of performance.

Although the controller was designed for the 100% load operating condition, similar results are obtained for the other simulations specified by Dixon and Pike (2005). In particular, Table 6.3 presents the integral of absolute error from the set–point for all four output variables, for a sine wave load disturbance at all three operating conditions, i.e. 0%, 50% and 100%. For comparative purposes, the equivalent values given in the paper by Al Seyab and Cao (2006) are also presented in Table 6.3.

Comparing the values, it is evident that the NMSS/MPC controller in general

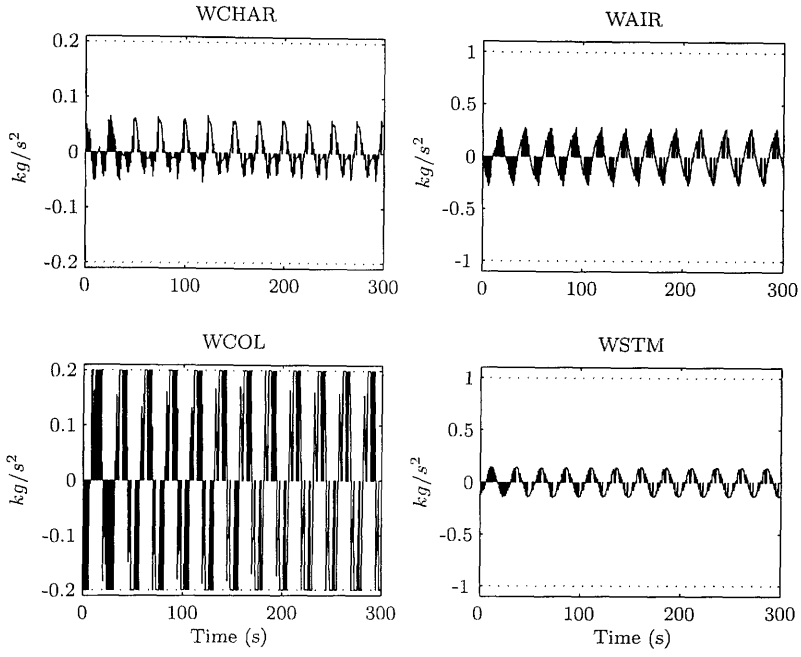

Figure 6.11: Rate–of–change for the four control input signals of the ALSTOM gasifier at 100% load using the forward path form.

performs better for the 100% and 50% load conditions (exept for the PGAS output variable where the controller of Al Seyab and Cao (2006) uses a non–linear model to estimare), while the nonlinear controller of Al Seyab and Cao (2006) is superior (for some variables) at the 0% load condition. Bearing in mind that the controller of Al Seyab and Cao (2006) is designed for the 0% load condition, while the one presented here is based on the 100% load case (as initially required by the benchmark organisers), this conclusion is to be expected. The fact that the NMSS/MPC controller yields better results for the middle 50% load condition suggests that it may be more robust to operating level changes. However, this result needs further investigation and is included here merely to show that the NMSS/MPC controller can produce at least similar results to existing MPC formulations.

Finally, it should be pointed out that the NMSS/MPC algorithm yields similar results to the NMSS/PIP controller also developed for the gasifier (Taylor and Shaban, 2006). However, the latter algorithm was obtained by an off–line trial and error tuning of the control weighting terms using the full nonlinear simulation,

in order to minimise activation of the constraints, while the new NMSS/MPC approach builds these constraints directly into the design process.

6.4 Conclusions

This chapter, motivated by earlier research on NMSS/MPC by Wang and Young (2006), in which the NMSS formulation is compared to the conventional minimal solution, has considered the implementation structure of this NMSS/MPC controller. In particular it has identified the controller of Wang and Young (2006) as a conventional feedback structure, and considering earlier research on NMSS/PIP control by Taylor et al. (1998) proposed an alternative arrangement that takes advantage of the identified model, namely the Forward Path NMSS/MPC controller.

The relative robustness of the proposed structures are evaluated by Monte Carlo Simulation (MCS). Here, the model parameters for each stochastic realisation are obtained from the joint probability distribution of the parameter estimates. With the availability of fast digital computers, this provides one of the simplest and most attractive approaches to assessing the sensitivity of a controller to parametric uncertainty and is widely applicable in the process control industry.

Such MCS analysis, including examination of the closed loop pole positions, suggests that the suggested Forward Path MPC scheme generally offers improved robustness properties in comparison to the feedback approach of Wang and Young (2006), whilst preserving the latter's disturbance rejection properties. Furthermore, the control signals are typically smoother in the forward path case. It is recommended, however, that the feedback structure is used when dealing with marginally stable or unstable systems. In these cases, the forward path controller is more likely to yield an unstable realisation in the presence of parametric uncertainty. These characteristics are of particular importance in real applications where system uncertainty is inevitable and smooth handling of the control signal actuators is nearly always desirable.

Furthermore, it was shown that the proposed control structure can produce at least similar (if not better) results compared to a nonlinear controller presented by Al Seyab and Cao (2006). Therefore, it is suggested that this particular structure can pose a simple alternative to more complicated nonlinear controllers.

Chapter 7

Disturbance Handling

7.1 Introduction

This chapter presents a NMSS/MPC formulation for systems with a modelled measured disturbance. Modelled disturbances can be any measurable but uncontrollable system inputs that affect the system state or output. Examples can include wind speed and environmental temperature in ventilation systems, environmental conditions (e.g. heavy rain) in wastewater treatment plants and others that are generally hard to control but relatively easy to measure. Although it could be argued that integrators in the control design will deal with any constant disturbances (see Muske and Badgwell, 2002; Pannocchia and Rawlings, 2003, for a detailed approach to non–modelled disturbances), it is true that in this context better predictions (that result from better modelling) will lead to better control performance (Rossiter, 2003).

In the relevant literature, two ways of dealing with modelled disturbances are considered (see Bemporad, 2006, for a brief review), namely improving the predictions by taking into account the disturbance measurement and model (Rossiter, 2003, Sec. 4.9.3) and augmenting the state vector with disturbance values (Maciejowski, 2002, Sec. 7.4). An implementation of the latter is also presented in Sanchez and Katebi (2003) for a wastewater treatment simulation.

Here, both approaches are presented for the NMSS/MPC framework. The system description is given and the optimisation problem is defined. A simulation example shows the improvement in the tracking performance of the presented strategies compared to approaches where the disturbance model is not taken into account. Finally, a case study of temperature control in an installation is presented, where the temperature of the surrounding environment is considered a disturbance.

7.2 System description

The methodological approach presented below follows from earlier research into model predictive control (Wang and Young, 2006) based on the representation of the system using non–minimal state space models (Young et al., 1987; Taylor et al., 2000a) with the addition of the modelled disturbance signals. The controller of Wang and Young (2006) is extended to take into account modelled disturbances (either by augmenting the state vector or explicitly). While integral action is still accounted for inherently by definition of the system in terms of the differenced input, output and disturbance signals. In this regard, consider the q–input, p–output system with r measured disturbance signals described by the following difference equation

$$\begin{aligned}\mathbf{y}_k = &- \mathbf{A}_1\mathbf{y}_{k-1} - \mathbf{A}_2\mathbf{y}_{k-2} - \ldots - \mathbf{A}_n\mathbf{y}_{k-n} \\ &+ \mathbf{B}_1\mathbf{u}_{k-1} + \mathbf{B}_2\mathbf{u}_{k-2} + \ldots + \mathbf{B}_m\mathbf{u}_{k-m} \\ &+ \mathbf{G}_1\mathbf{v}_{k-1} + \mathbf{G}_2\mathbf{v}_{k-2} + \ldots + \mathbf{G}_l\mathbf{v}_{k-l}\end{aligned}$$

where $\mathbf{y}_k = [y_{1,k} \quad y_{2,k} \quad \cdots \quad y_{p,k}]^T$, $\mathbf{u}_k = [u_{1,k} \quad u_{2,k} \quad \cdots \quad u_{q,k}]^T$ and $\mathbf{v}_k = [v_{1,k} \quad v_{2,k} \quad \cdots \quad v_{r,k}]^T$ are the output, the control input and the disturbance vectors[1] respectively, and the $\mathbf{A}_i$, $\mathbf{B}_j$ and $\mathbf{G}_k$ are $p \times p$, $p \times q$ and $p \times r$ matrices

[1] Although $\mathbf{v}$ is a vector of disturbance signals, in terms of the estimation algorithm it can be treated as any other input of the system. In this regard, the SRIV algorithm is utilised to determine the value of l and the matrices $\mathbf{G_k}$ as described in Section 2.1.

respectively, for $i = 1, \ldots, n$, $j = 1, \ldots, m$ and $k = 1, \ldots, l$.

The above system can represented in terms of the differenced output, input and disturbance variables, yielding embedded integral action as

$$\begin{aligned}\Delta \mathbf{y}_k = &- \mathbf{A}_1 \Delta \mathbf{y}_{k-1} - \mathbf{A}_2 \Delta \mathbf{y}_{k-2} - \ldots - \mathbf{A}_n \Delta \mathbf{y}_{k-n} \\ &+ \mathbf{B}_1 \Delta \mathbf{u}_{k-1} + \mathbf{B}_2 \Delta \mathbf{u}_{k-2} + \ldots + \mathbf{B}_m \Delta \mathbf{u}_{k-m} \\ &+ \mathbf{G}_1 \Delta \mathbf{v}_{k-1} + \mathbf{G}_2 \Delta \mathbf{v}_{k-2} + \ldots + \mathbf{G}_l \Delta \mathbf{v}_{k-l}\end{aligned} \tag{7.1}$$

Based on the difference equation (7.1) the following sections describe two different approaches to disturbance handling, namely by augmenting the state vector (Section 7.2.1) and by direct evaluation of the disturbance model in the prediction equations (Section 7.2.2).

7.2.1 Augmenting the state vector

Based on the state space representation of Wang and Young (2006) the NMSS state vector (4.2) is augmented with disturbance measurements as:

$$\begin{aligned}\Delta \mathbf{x}_{\Delta,k}^T = \big[\Delta \mathbf{y}_k^T \quad \Delta \mathbf{y}_{k-1}^T \quad \cdots \quad \Delta \mathbf{y}_{k-n+1}^T \quad \Delta \mathbf{u}_{k-1}^T \quad \Delta \mathbf{u}_{k-2}^T \quad \cdots \quad \Delta \mathbf{u}_{k-m+1}^T \\ \Delta \mathbf{v}_k^T \quad \Delta \mathbf{v}_{k-1}^T \quad \cdots \quad \Delta \mathbf{v}_{k-l+1}^T\big]\end{aligned}$$

The NMSS state space form is now

$$\Delta \mathbf{x}_{\Delta,k+1} = \mathbf{A}_\Delta \Delta \mathbf{x}_{\Delta,k} + \mathbf{B}_\Delta \Delta \mathbf{u}_k + \mathbf{G}_\Delta \Delta \mathbf{v}_k \tag{7.2a}$$

$$\Delta \mathbf{y}_{k+1} = \mathbf{C}_\Delta \Delta \mathbf{x}_{\Delta,k+1} \tag{7.2b}$$

where the system matrices can be partitioned as

$$\mathbf{A}_\Delta = \begin{bmatrix} \mathbf{A}_{\Delta,1} & \mathbf{A}_{\Delta,2} \\ \mathbf{0}_l & \mathbf{A}_{\Delta,3} \end{bmatrix}; \mathbf{B}_\Delta = \begin{bmatrix} \mathbf{B}_{\Delta,1} \\ \mathbf{0}_{lq} \end{bmatrix}; \quad \mathbf{G}_\Delta = \begin{bmatrix} \mathbf{0}_{m,1} \\ \mathbf{G}_{\Delta,1} \end{bmatrix}; \mathbf{C}_\Delta = \begin{bmatrix} \mathbf{C}_{\Delta,1} & \mathbf{0}_{pl} \end{bmatrix}$$

Table 7.1: Dimensions of the matrices in the state space form

Matrix	Dimension
$\mathbf{A}_{\Delta,1}$	$np+(m-1)q \times np+(m-1)q$
$\mathbf{A}_{\Delta,2}$	$np+(m-1)q \times lr$
$\mathbf{A}_{\Delta,3}$	$lr \times lr$
$\mathbf{B}_{\Delta,1}$	$np+(m-1)q \times q$
$\mathbf{G}_{\Delta,1}$	$np+(m-1)q \times r$
$\mathbf{C}_{\Delta,1}$	$p \times np+(m-1)q$

The $\mathbf{A}_{\Delta,1}$, $\mathbf{B}_{\Delta,1}$ and $\mathbf{C}_{\Delta,1}$ are given by (4.3) and $\mathbf{A}_{\Delta,2}$, $\mathbf{A}_{\Delta,3}$ and $\mathbf{G}_{\Delta,1}$ are defined as:

$$\mathbf{A}_{\Delta,2} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{G}_{l-1} & \mathbf{G}_l \\ \mathbf{0}_{pr} & \mathbf{0}_{pr} & \cdots & \mathbf{0}_{pr} & \mathbf{0}_{pr} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_{pr} & \mathbf{0}_{pr} & \cdots & \mathbf{0}_{pr} & \mathbf{0}_{pr} \end{bmatrix} ; \mathbf{A}_{\Delta,3} = \begin{bmatrix} \mathbf{0}_r & \mathbf{0}_r & \cdots & \mathbf{0}_r & \mathbf{0}_r \\ \mathbf{I}_r & \mathbf{0}_r & \cdots & \mathbf{0}_r & \mathbf{0}_r \\ \mathbf{0}_r & \mathbf{I}_r & \cdots & \mathbf{0}_r & \mathbf{0}_r \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0}_r & \mathbf{0}_r & \cdots & \mathbf{I}_r & \mathbf{0}_r \end{bmatrix}$$

and

$$\mathbf{G}_{\Delta,1}^T = \begin{bmatrix} \mathbf{I}_r^T & \mathbf{0}_r & \cdots & \mathbf{0}_r & \mathbf{0}_r & \mathbf{0}_r & \mathbf{0}_r & \cdots & \mathbf{0}_r & \mathbf{0}_r \end{bmatrix}$$

while $\mathbf{0}_{m,1}$ is a $np+(m-1)q \times l$ matrix of zeros. The dimensions of the above matrices are given in Table 7.1.

The state vector is subsequently redefined, to include the current measurement of the system outputs as:

$$\mathbf{x}_k^T = \begin{bmatrix} \Delta\mathbf{x}_{\Delta,k}^T & \mathbf{y}_k^T \end{bmatrix}$$

It should be noted that the state vector still consists of only measured system variables and there is no need for a state reconstructor. In the same manner as described in Section 4.1, the state space description of the system takes the form:

$$\begin{bmatrix} \Delta \mathbf{x}_{\Delta,k+1} \\ \mathbf{y}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_\Delta & \mathbf{0} \\ \mathbf{C}_\Delta \mathbf{A}_\Delta & \mathbf{I}_p \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_{\Delta,k} \\ \mathbf{y}_k \end{bmatrix} + \begin{bmatrix} \mathbf{B}_\Delta \\ \mathbf{C}_\Delta \mathbf{B}_\Delta \end{bmatrix} \Delta \mathbf{u}_k + \begin{bmatrix} \mathbf{G}_\Delta \\ \mathbf{C}_\Delta \mathbf{G}_\Delta \end{bmatrix} \Delta \mathbf{v}_k \tag{7.3a}$$

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{0}^T & \mathbf{I}_p \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_{\Delta,k} \\ \mathbf{y}_k \end{bmatrix} \tag{7.3b}$$

where $\mathbf{x}$ is of dimension $(n+1)p+(m-1)q+lr \times 1$ and $\mathbf{0}$ is a $(n+1)p+(m-1)q+lr \times p$ zero matrix. The system is therefore described in the following state space form

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\Delta \mathbf{u}_k + \mathbf{G}\Delta \mathbf{v}_k \tag{7.4a}$$

$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1} \tag{7.4b}$$

where the system matrices are defined in (7.3).

7.2.2 Direct use of the disturbance model

In this section the state space model that is used for the second approach to disturbance handling is developed. In this case, the state vector is the one of Wang and Young (2006) given by (4.4), while the state space description is extended to take into account the disturbance model.

Consider again a system model in the difference equation form of (7.1). To describe the system in the a NMSS form similar to the one of Wang and Young (2006), the state vector is defined as in (4.4) and the corresponding state space model is defined as,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\Delta \mathbf{u}_k + \mathbf{G}\Delta \bar{\mathbf{v}}_k \tag{7.5a}$$

$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1} \tag{7.5b}$$

in which

$$\Delta\bar{\mathbf{v}}_k = \begin{bmatrix} \mathbf{v}_k - \mathbf{v}_{k-1} \\ \mathbf{v}_{k-1} - \mathbf{v}_{k-2} \\ \vdots \\ \mathbf{v}_{k-l+1} - \mathbf{v}_{k-l} \end{bmatrix}$$

and the system matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are described as in (4.6), while:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{G}_l \\ \mathbf{0}_{pr} & \mathbf{0}_{pr} & \cdots & \mathbf{0}_{pr} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{pr} & \mathbf{0}_{pr} & \cdots & \mathbf{0}_{pr} \\ \mathbf{0}_{qr} & \mathbf{0}_{qr} & \cdots & \mathbf{0}_{qr} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{qr} & \mathbf{0}_{qr} & \cdots & \mathbf{0}_{qr} \\ \mathbf{G}_1 & \mathbf{G}_2 & \cdots & \mathbf{G}_l \end{bmatrix}$$

Remark 7.2.1. *From the above, it is clear that, in comparison to the second approach, augmenting the state space model increases the state dimension. Depending on the disturbance model this increase can be considerable compared to the initial state dimension. However, as pointed out in Bemporad (2006) it is a more general approach to rejecting measured disturbances in control systems that are based on linear models and therefore it is considered here for completeness (an application of this approach to PIP control of systems with measured modelled disturbances can be found in Lees (1996)).*

7.3 Control Description

In both the above system descriptions, the controller is derived from the numerical minimisation of the index[2] (6.2) at each sampling instant, subject to the system constraints (4.12b)

In the following, the above optimisation problem is expressed by a more compact matrix form where the system predictions are based on the system description (7.5). Clearly, by substitution of $\boldsymbol{\Delta}\bar{\mathbf{v}}$ with $\boldsymbol{\Delta}\mathbf{v}$ and using the appropriate matrices, the same result holds for the system description (7.4) that is based on an augmented state vector.

In a similar manner to Section 4.2 and assuming that the control action retains it's last value after the end of the control horizon (i.e. $\boldsymbol{\Delta}\mathbf{u}_{k+i} = \mathbf{0}$, $i = N_c, \ldots, N_p$), the system prediction equations can be formed as:

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\boldsymbol{\Delta}\mathbf{u}_k + \mathbf{G}\boldsymbol{\Delta}\bar{\mathbf{v}}_k \\
\mathbf{x}_{k+2} &= \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\boldsymbol{\Delta}\mathbf{u}_{k+1} + \mathbf{G}\boldsymbol{\Delta}\bar{\mathbf{v}}_{k+1} \\
&= \mathbf{A}^2\mathbf{x}_k + (\mathbf{A}\mathbf{B}\boldsymbol{\Delta}\mathbf{u}_k + \mathbf{B}\boldsymbol{\Delta}\mathbf{u}_{k+1}) + (\mathbf{A}\mathbf{G}\boldsymbol{\Delta}\bar{\mathbf{v}}_k + \mathbf{G}\bar{\mathbf{v}}_{k+1}) \\
&\vdots \\
\mathbf{x}_{k+N_c} &= \mathbf{A}^{N_c}\mathbf{x}_k + \sum_{i=0}^{N_c-1} \mathbf{A}^i\mathbf{B}\boldsymbol{\Delta}\mathbf{u}_{k+N_c-1-i} + \sum_{i=0}^{N_c-1} \mathbf{A}^i\mathbf{G}\boldsymbol{\Delta}\bar{\mathbf{r}}_{k+N_c-1-i} \\
&\vdots \\
\mathbf{x}_{k+N_p} &= \mathbf{A}^{N_p}\mathbf{x}_k + \sum_{i=N_p-N_c}^{N_p-1} \mathbf{A}^i\mathbf{B}\boldsymbol{\Delta}\mathbf{u}_{k+N_p-1-i} + \sum_{i=0}^{N_p-1} \mathbf{A}^i\mathbf{G}\boldsymbol{\Delta}\bar{\mathbf{v}}_{k+N_p-1-i}
\end{aligned}
$$

Remark 7.3.1. *In the above prediction equations it is assumed that future values of the rate-of-change (and therefore their actual values) for the disturbance signals are available. However, in most real situations they are not available and a decision should be made regarding the predictions. There are numerous alternatives, based on both the designers' knowledge on the nature of the disturbance and the*

[2]This index is identical to the one used by Wang and Young (2006) and in Chapter 6. However, the output predictions are based on the models described in the previous sections.

computational load that can be handled online. To name a few, the evolution of the disturbance can take the form of a predefined model (e.g. an AR model) that can be used for predicting future values, or simply the disturbance can be constrained to retain its' present value in the future. Another widely used approach is to consider the disturbance signal as the output of a predefined model driven by white noise.

From this point on, the disturbance in the prediction equations is treated following the next assumption.

Assumption 7.3.1. *The disturbance signals retain their present value throughout the prediction horizon (i.e.* $\mathbf{v}_{k+i} = \mathbf{v}_k$, $i = 1, \ldots, N_p$*)*

Subsequently, the future state, output, and control increment vectors are given by (4.7), the future set-point trajectory vector by (4.9) and the future control signal vector by (4.13). From Assumption 7.3.1 it follows that the $\boldsymbol{\Delta}\bar{\mathbf{v}}$ vanishes to a zero vector after l samples into the future (i.e. $\boldsymbol{\Delta}\bar{\mathbf{v}}_{k+i} = \mathbf{0}$ for $i = l+1, \ldots, N_p$). In this regard, the future disturbance vector is defined as[3]:

$$\Delta\mathbf{V}^T = \begin{bmatrix} \Delta\bar{\mathbf{v}}_k^T & \Delta\bar{\mathbf{v}}_{k+1}^T & \ldots & \Delta\bar{\mathbf{v}}_{k+l}^T \end{bmatrix}$$

The state and output prediction equations can now take the form:

$$\mathbf{X} = \mathbf{F}\mathbf{x}(k) + \Phi\boldsymbol{\Delta}\mathbf{U} + \mathbf{H}_v\boldsymbol{\Delta}\mathbf{V} \tag{7.6a}$$

$$\mathbf{Y} = \bar{\mathbf{C}}\mathbf{X} \tag{7.6b}$$

[3]In the general case where the evolution of the disturbance is considered known or modelled, $\Delta\mathbf{V}$ could be defined as $\Delta\mathbf{V}^T = \begin{bmatrix} \Delta\bar{\mathbf{v}}_k^T & \Delta\bar{\mathbf{v}}_{k+1}^T & \ldots & \Delta\bar{\mathbf{v}}_{k+N_p-1}^T \end{bmatrix}$ and the $\mathbf{H}_v$ would have the appropriate dimensions. However, in the present this would only increase the dimensions of the related matrices without providing more information on the disturbance values (they would be zero).

where $\mathbf{F}$ and $\mathbf{\Phi}$ are given by (4.8), $\mathbf{H}_v$ is defined as

$$\mathbf{H}_v = \begin{bmatrix} \mathbf{G} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AG} & \mathbf{G} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{l}\mathbf{G} & \mathbf{A}^{l-1}\mathbf{G} & \cdots & \mathbf{G} \\ \mathbf{A}^{l+1}\mathbf{G} & \mathbf{A}^{l+2}\mathbf{G} & \cdots & \mathbf{AG} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}^{N_p-1}\mathbf{G} & \mathbf{A}^{N_p-2}\mathbf{G} & \cdots & \mathbf{A}^{N_p-l-1}\mathbf{G} \end{bmatrix}$$

and $\bar{\mathbf{C}}$ is a N_p block diagonal matrix of dimensions $pN_p \times ((n+1)p+(m-1)q)N_p$, with the matrix $\mathbf{C}$ on it's diagonal. The optimisation problem presented at the beginning of this section can then take the general form of the MPC problem (2.4):

$$\min_{\Delta \mathbf{U}} J_s = (\mathbf{S} - \mathbf{Y})^T \bar{Q} (\mathbf{S} - \mathbf{Y}) + \Delta \mathbf{U}^T \bar{R} \Delta \mathbf{U} \tag{7.7a}$$

subject to,

$$\begin{cases} -\underline{\Delta \mathbf{U}} \leq \Delta \mathbf{U} \leq \overline{\Delta \mathbf{U}} \\ -\underline{\mathbf{U}} \leq \mathbf{U} \leq \overline{\mathbf{U}} \\ -\underline{\mathbf{Y}} \leq \mathbf{Y} \leq \overline{\mathbf{Y}} \end{cases} \tag{7.7b}$$

where the notation $\underline{\cdot}$ and $\overline{\cdot}$ is used to denote lower and upper boundaries for the appropriate vectors; the inequalities refer to element by element inequalities and the $\bar{\mathbf{Q}}$ and $\bar{\mathbf{R}}$ matrices are defined as,

$$\bar{\mathbf{Q}} = diag \underbrace{\begin{bmatrix} \mathbf{Q} & \mathbf{Q} & \cdots & \mathbf{Q} \end{bmatrix}}_{N_p}$$

$$\bar{\mathbf{R}} = diag \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{R} & \cdots & \mathbf{R} \end{bmatrix}}_{N_c}$$

By straightforward manipulation of (7.6) and substitution on (7.7a), the optimi-

sation problem takes the form,

$$\begin{aligned}\min_{\Delta U} \ & \frac{1}{2}\Delta\mathbf{U}^T\left(\Phi^T\bar{\mathbf{C}}^T\bar{\mathbf{Q}}\bar{\mathbf{C}}\Phi+\bar{\mathbf{R}}\right)\Delta\mathbf{U}+ \\ & +\left[\Phi^T\bar{\mathbf{C}}\bar{\mathbf{Q}}\left(\bar{\mathbf{C}}\mathbf{F}\mathbf{x}(k)+\bar{\mathbf{C}}\mathbf{H}_v\Delta\mathbf{V}-\mathbf{S}\right)\right]^T\Delta\mathbf{U} \\ \text{subject to: } & \mathbf{M}\Delta\mathbf{U}\leq\mathbf{N}\end{aligned}$$

where the inequality $\mathbf{M}\Delta\mathbf{U} \leq \mathbf{N}$ is the combination of (7.7b) into a single inequality in terms of the decision variable $\Delta\mathbf{U}$. The $\mathbf{M}$ and $\mathbf{N}$ matrices are given by:

$$\mathbf{M}=\begin{bmatrix}-\mathbf{I}_{N_c}\\ \mathbf{I}_{N_c}\\ -\mathbf{C}_2\\ \mathbf{C}_2\\ -\bar{\mathbf{C}}\Phi\\ \bar{\mathbf{C}}\Phi\end{bmatrix};\ \mathbf{N}=\begin{bmatrix}-\underline{\Delta\mathbf{U}}\\ \overline{\Delta\mathbf{U}}\\ -\underline{\mathbf{U}}+\mathbf{C}_1u_{k-1}\\ \overline{\mathbf{U}}-\mathbf{C}_1u_{k-1}\\ -\underline{\mathbf{Y}}+\bar{\mathbf{C}}\mathbf{F}\mathbf{x}_k+\bar{\mathbf{C}}\mathbf{H}_v\Delta\mathbf{V}\\ \overline{\mathbf{Y}}-\bar{\mathbf{C}}\mathbf{F}\mathbf{x}_k-\bar{\mathbf{C}}\mathbf{H}_v\Delta\mathbf{V}\end{bmatrix}$$

and the $\mathbf{C}_1$ and $\mathbf{C}_2$ matrices are defined as follows,

$$\mathbf{C}_1=\begin{bmatrix}\mathbf{I}_p\\ \mathbf{I}_p\\ \vdots\\ \mathbf{I}_p\\ \mathbf{I}_p\end{bmatrix};\ \mathbf{C}_2=\begin{bmatrix}\mathbf{I}_p & \mathbf{0}_p & \cdots & \mathbf{0}_p & \mathbf{0}_p\\ \mathbf{I}_p & \mathbf{I}_p & \cdots & \mathbf{0}_p & \mathbf{0}_p\\ \vdots & \vdots & \ddots & & \vdots\\ \mathbf{I}_p & \mathbf{I}_p & \cdots & \mathbf{I}_p & \mathbf{0}_p\\ \mathbf{I}_p & \mathbf{I}_p & \cdots & \mathbf{I}_p & \mathbf{I}_p\end{bmatrix}$$

7.4 Simulation Examples

In the following, two simulation examples are considered. Initially, a SISO second order discrete time system with modelled measured disturbance is considered to highlight the differences among control structures that take into account the

disturbance model and the ones that don't. Then, a MIMO temperature control installation is considered where the modelling and control simulation procedures are presented for a complete approach to control a real installation with modelled measured disturbance. In this latter example, only the approach presented in Section 7.2.2 is considered because it does not increase the size of the optimisation problem and is more suitable for the outdated hardware that accompanies the installation.

7.4.1 SISO Example

Consider the second order discrete time system with modelled measured disturbance described by the following difference equation[4]:

$$y_k - 1.4y_{k-1} + 0.45y_{k-2} = 0.3u_{k-1} - 0.27u_{k-2} + 0.4v_{k-1} - 0.2v_{k-2}$$

where y is the system output, u is the control input and v is a measured disturbance. It is straightforward to express the above model in the form of (7.1) that is the basis for both system descriptions considered in this chapter. For comparative purposes, in the following, a NMSS/MPC controller is evaluated for both the system with the augmented state and that which models the disturbance directly. Their performance is subsequently compared to the alternative control structure presented in Chapter 4 and the controller of Wang and Young (2006) that do not take into account the disturbance model. In these latter cases it is assumed that a measurement of the disturbance is not available. Although such a comparison is in favor of the structures that make use of the measured disturbance, it is merely performed to show that making use of the available measurement can dramatically improve performance.

[4]This system is introduced by Lees (1996) and is used here for a possible comparison between the different control structures.

7.4.1.1 Augmenting the state vector

By the description presented in Section 7.2.1 the augmented state vector for the above system is:

$$\mathbf{x}_k = \begin{bmatrix} \Delta y_k & \Delta y_{k-1} & \Delta u_{k-1} & \Delta v_k & \Delta v_{k-1} & y_k \end{bmatrix}^T$$

The state space form is subsequently defined as in (7.4) with the system matrices and vectors as described in Section 7.2.1

7.4.1.2 Direct use of the disturbance model

To describe the system in the form presented in Section 7.2.2 for direct use of the disturbance model in the prediction equations, the state vector is defined as:

$$\mathbf{x}_k = \begin{bmatrix} \Delta y_k & \Delta y_{k-1} & \Delta u_{k-1} & y_k \end{bmatrix}^T$$

Subsequently, the system takes the form (7.5) with the system matrices and vectors defined in as Section 7.2.2 while the difference of the disturbance vector $\Delta\bar{\mathbf{v}}$ is given by:

$$\Delta\bar{\mathbf{v}} = \begin{bmatrix} v_k - v_{k-1} \\ v_{k-1} - v_{k-2} \end{bmatrix}$$

As already mentioned in Remark 7.2.1 it is clear from the above system descriptions that augmenting the state vector to account for the disturbance model increases the dimension of the system description. However, the present chapter presents both approaches and demonstrates their application to NMSS/MPC without attempting to compare them directly.

7.4.1.3 Control Simulation

For the purposes of the present example, the system is simulated in closed loop with the desired reference signal considered to be a step from zero to one units

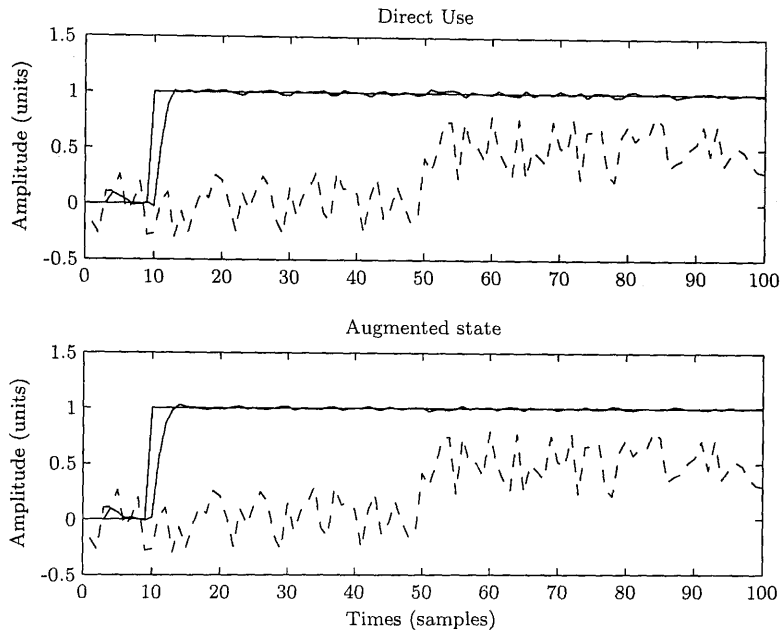

Figure 7.1: Reference signal (thick solid lines), System output (thin solid lines) and disturbance signal (dashed lines) for a simulation example for the control structure that directly accounts for the disturbance model (top) and the one based on an augmented system model (bottom).

and occurring after 10 samples. The disturbance signal is defined as white noise with standard deviation $\sigma^2 = 0.3$ units and it's mean value is changing from zero to 0.5 units after 50 samples. The resulting simulations are presented in Figure 7.1 for the two control structures introduced here, and in Figure 7.2 for the controller presented in Chapter 4 and the controller of Wang and Young (2006). It should be noted the the latter two do not take into account the disturbance model in the prediction equations. Visual inspection of Figures 7.1 and 7.2 shows that the control structures that account for the disturbance model, reject the disturbance almost completely despite the relatively large amplitude of the disturbance signal.

For a more detailed look, Table 7.2 presents the absolute error at every sampling instant for each one of the control structures. The improved results when the disturbance model is taken into account are evident and supports the fact that the better the model, the better the control when MPC is considered. In addition, for the present example, the controller that is based in the system with

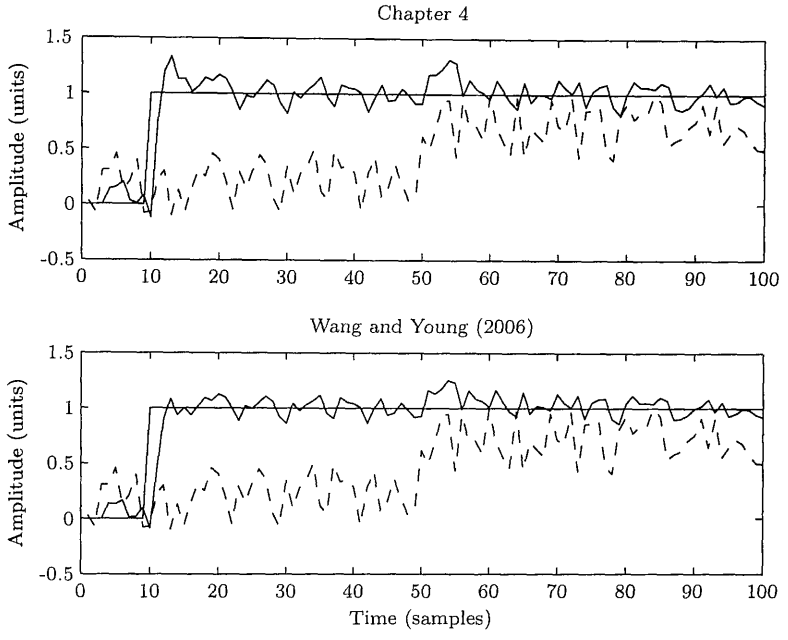

Figure 7.2: Reference signal (thick solid lines), System output (thin solid lines) and disturbance signal (dashed lines) for a simulation example for the control structure presented in Chapter 4 (top) and the controller of Wang and Young (2006) (bottom).

an augmented state produces better results in terms of disturbance rejection than the one that explicitly takes into account the disturbance model in the prediction equations. Furthermore, it should be noted that all controllers reject a constant disturbance (the change in the mean value of the disturbance signal in this case), which is expected as this is already considered in Section 4.3 and the paper of Wang and Young (2006). However, the effect of the constantly varying part of the disturbance signal (the white noise in this case) is evident in the cases of the controllers that don't account for the disturbance model and considerably reduced for the structures presented in this chapter.

7.4.2 Control of a temperature control installation

In the following, a simulation of NMSS/MPC control using the system description presented in Section 7.2.2 that explicitly accounts for the disturbance model is shown.

Table 7.2: Sum of absolute errors for the controller that makes direct use of the disturbance signal, the controller that is based on an augmented system model, the controller presented in Chapter 4 and the controller of Wang and Young (2006).

Controller	Sum of absolute errors
Direct Use	3.0156
Augmented state	2.5159
Chapter 4	9.7692
Wang and Young (2006)	8.6010

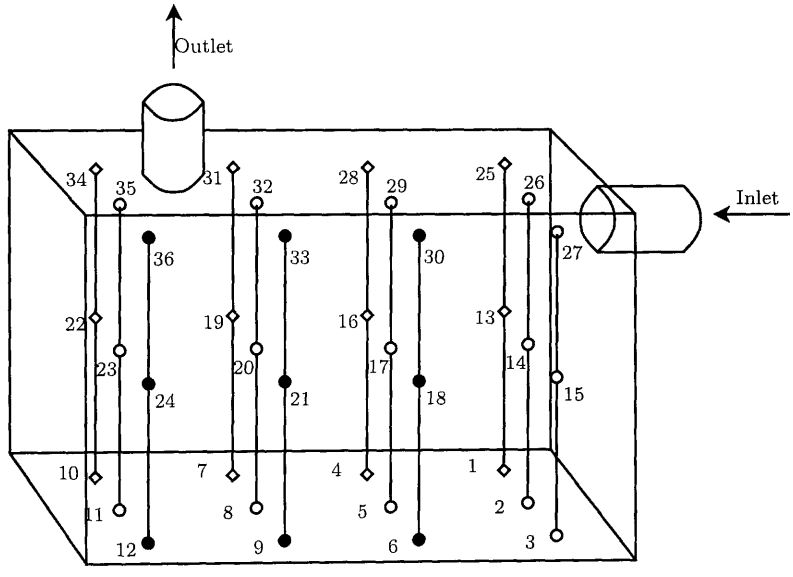

Figure 7.3: Schematic of the installation. The places of the temperature actuators are also shown, with different markers denoting the three areas that were identified as having a different response to the controlled inputs and disturbance.

7.4.2.1 Description of the installation

This section presents a brief description of an installation for temperature control, located in Leuven (Belgium)[5], a schematic of which is presented in Figure 7.3. The installation has dimensions 12m long × 4m wide × 5m high and is located near the wall of a large room. Temperature sensors (type–T thermocouples) are located inside the installation as depicted in Figure 7.3 while a measurement of the surrounding temperature is also available. The manipulated variables are the inlet air temperature and ventilation rate, while the surrounding temperature outside

[5]The data acquisition was performed by members of the M3–BIORES group of the Catholic University of Leuven in Belgium. Thanks goes to everyone involved and especially Prof. Daniel Berckmans for agreeing to use this dataset in this thesis.

the installation can be considered as a disturbance signal.

Experimentation and evaluation of the step experiment results presented in the following section (Figure 7.4) suggested that the temperature sensors inside the installation can be divided into three groups (marked with different markers in Figure 7.3) depending on the way they respond to changes in the control inputs and the disturbance. The first group (Group I) is located near the 'front' wall of the installation (marked with • in Figure 7.3) and is greatly affected by the surrounding temperature, requiring higher effort from the controller to alter their temperature. The second group (Group II) is comprised by the sensors that are located near the 'back' wall of the installation (marked with ◇ in Figure 7.3) that are also greatly affected by the surrounding temperature but, on average, maintain a higher temperature than those of Group I. Finally, the third group (Group III) includes the actuators in the 'middle' part of the installation and the actuators with numbers 3, 15 and 27 (marked with ○ in Figure 7.3). The temperature at this group seems to be easier to control and responds faster to the control inputs than the two other groups.

The control objective is to maintain the temperature at every point inside the installation as close as possible to 23°C. However, the size of the installation and the placement of the fans result in an uneven spread of the heat inside the installation. Therefore, there is a spread in the temperature at different places inside the installation. In order to minimise the variation of the temperature, the average temperature of Group I and the average temperature of Group II are controlled. These two, somehow conflicting objectives (in general, Group II has higher temperature than Group I), provide a reasonable value of the temperature variation throughout the installation. The output, input and disturbance signals are summarised in Table 7.3.

Table 7.3: Input, Output and Disturbance signals

Signal	Description
y_1	Average temperature of Group I (marked with $\diamondsuit$ in Fig. 7.3)
y_2	Average temperature of Group II (marked with $\bullet$ in Fig. 7.3)
u_1	Ventilation rate
u_2	Inlet temperature
v	Surrounding temperature

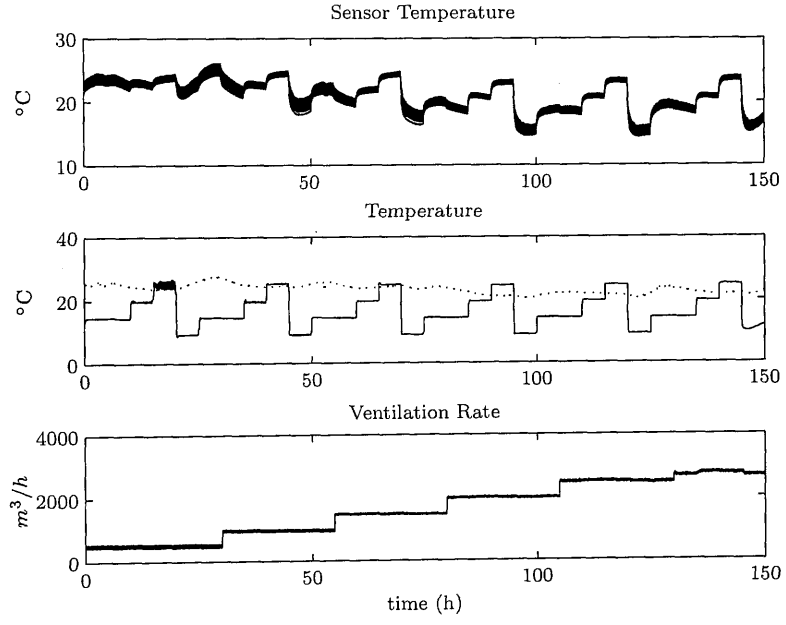

Figure 7.4: Step Experiment. Temperature response at every sensor (top figure); Inlet (solid line) and surrounding (dotted line) temperature (middle figure); Ventilation rate (bottom figure).

7.4.2.2 System Identification

As explained in Section 2.1, in order to estimate a linearised model for the system, a step experiment was conducted to the installation during a 6 day period in October 2006 and data were collected with a sampling rate of 1 sample every 6 minutes that is faster than 10 times the time constant of the system. The experiment included steps in both the ventilation rate and the temperature of the incoming air while at the same time the temperature of the surrounding environment (disturbance) was measured. The result for all 36 actuators is presented in Figure 7.4.

Experimentation suggested that using $n = 1$, $m = 3$ and $l = 3$ (as defined in

(7.1)) results in an accurate ($R_T^2 = 0.95$) yet robust ($YIC = -7.6$) description of the system. The average temperature of Group I and Group II are subsequently modelled for control purposes resulting in the system description of (7.1) with the following parameters:

$$\mathbf{A}_1 = \begin{bmatrix} -0.9799 & 0 \\ 0 & -0.9788 \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} -0.0011 & 0 \\ -0.0009 & 0 \end{bmatrix}; \mathbf{B}_2 = \begin{bmatrix} 0 & 0.0596 \\ 0 & 0.1201 \end{bmatrix}; \mathbf{B}_3 = \begin{bmatrix} 0 & -0.0517 \\ 0 & -0.1105 \end{bmatrix}$$

and

$$\mathbf{G}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \mathbf{G}_2 = \begin{bmatrix} 0.8952 \\ 0.8288 \end{bmatrix}; \mathbf{G}_3 = \begin{bmatrix} -0.8818 \\ -0.8162 \end{bmatrix}$$

In order to simulate the real conditions more accurately, every actuator response is individually modelled allowing for the n, m and l parameters to vary for better model fit.

It should be noted here that the lowest and highest ventilation rates are not used during the identification process since in the former case the air flow in the installation was not high enough and the temperature response is not very clear and in the latter the ventilation rate is not steady (probably because the low level controller that controls the ventilation rate does not respond well in this area). The model fit in each case (for the whole period of 150h) is presented in Figure 7.5.

7.4.2.3 Control and simulation results

For the present simulation, the control and prediction horizons are chosen $N_p = 10$ and $N_c = 3$ respectively, while the weighting matrices are $\mathbf{Q} = diag[1 \quad 1]$ and $\mathbf{R} = diag[1 \quad 1]$. The boundary values and the rate-of-change constraints for the

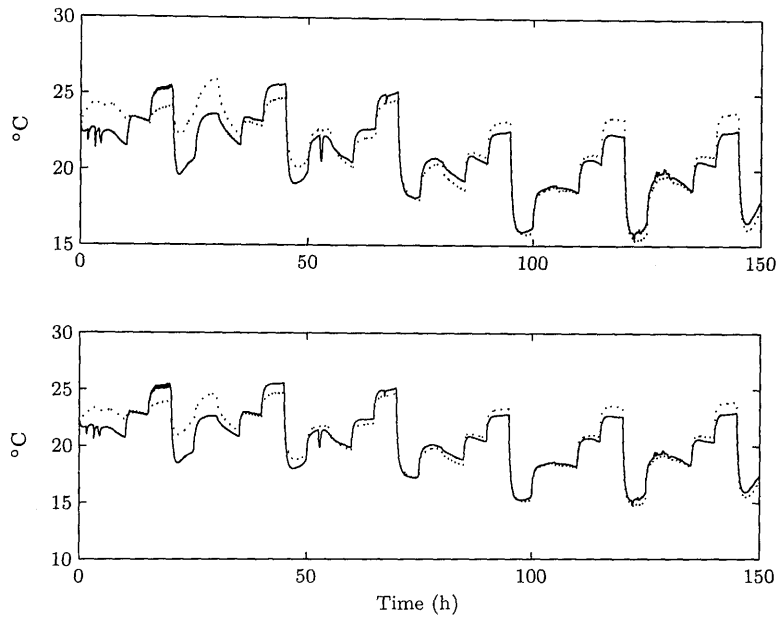

Figure 7.5: Actual (dotted lines) and estimated (solid lines) of the average temperature of Group I (top figure) and Group II (bottom figure).

Table 7.4: Input and rate–of–change bounds

Signal	Lower value	Upper Value
u_1	1000	2500
Δu_1	-350	350
u_2	15	27
Δu_2	-3	1

control inputs are derived by the acquired data and are presented in Table 7.4.

The system is subsequently simulated with the surrounding temperature having the same values as it had during the data acquisition. Clearly, any other choice of the disturbance temperature can be chosen, but care should be taken so that a realistic disturbance signal is selected (e.g. it should have a realistic rate–of–change and the necessary cyclic component between day and night–time) The resulting outputs along with the surrounding temperature are presented in Figure 7.6. It should be noted here that the results plotted in Figure 7.6 reflect the actual average temperature for each Group as it results from simulating every actuator separately and averaging their outputs and not from direct application of the estimated group models (this is only used to form the controller predictions). The simulated aver-

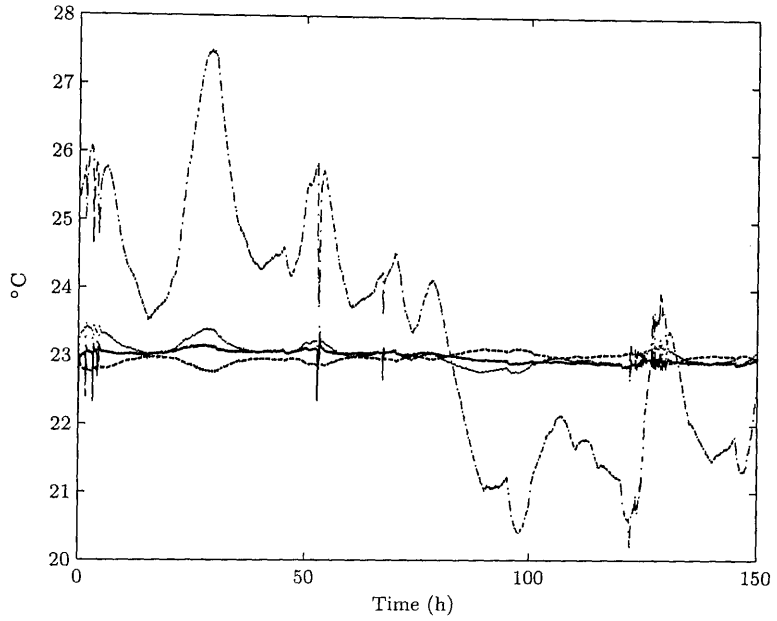

Figure 7.6: Average temperature of Group I (dotted line), Group II (dashed line), Group III (solid line) and the surrounding temperature (dash-dotted line).

ages of all Groups are kept near the target temperature of 23°C, despite the high variation (7°C) of the surrounding temperature. For completeness, the controlled variables are depicted in Figure 7.7.

Although the average temperatures are close to the target, it is equally important to follow the temperature at every point inside the installation. In this regard, the mean and maximum absolute errors from the target temperature are depicted in Figure 7.8. From this, it can be observed that despite the considerable variation of the surrounding temperature, the temperature inside the installation is maintained within 1°C at all times with an average deviation of 0.11°C. Even when there is a rapid decrease in the surrounding temperature of about 2°C in less than an hour at about the 55th hour (due to the opening of a large door of the room in which the installation is located) the simulated results show that the average error from the set-point is about 0.5°C.

For comparative purposes, the controller of Wang and Young (2006) is also applied to the system. Although the overall results seem to be similar to the ones using the controller presented here and are omitted here for brevity, there is a

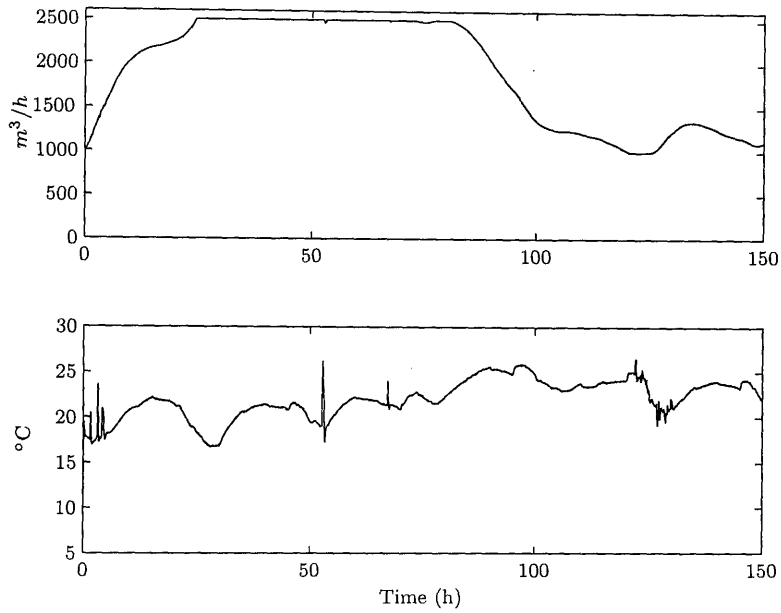

Figure 7.7: System Inputs. Ventilation rate (top figure); Inlet temperature (bottom figure)

difference in the handling of the manipulated variables. To highlight this difference, Figure 7.9 presents the rate–of–change for the inlet temperature variable. It is clear that, the controller presented in this chapter that accounts for the estimated disturbance model results in a smoother control signal. However, it is not clear whether there is going to be a difference in practical terms since the requested control input is not necessarily the same as the applied one (by observation of the step response data, the controlled variables don't have their exact nominal values which is probably due to the controllers responsible for driving the heater and the fan of the installation). Still, as already mentioned before, it is preferable to use the proposed controller that takes advantage of the modelled disturbance and produces smoother control signals.

7.5 Conclusion

This chapter presented a procedure for disturbance handling in the considered NMSS/MPC framework. Two approaches were considered, namely augmenting

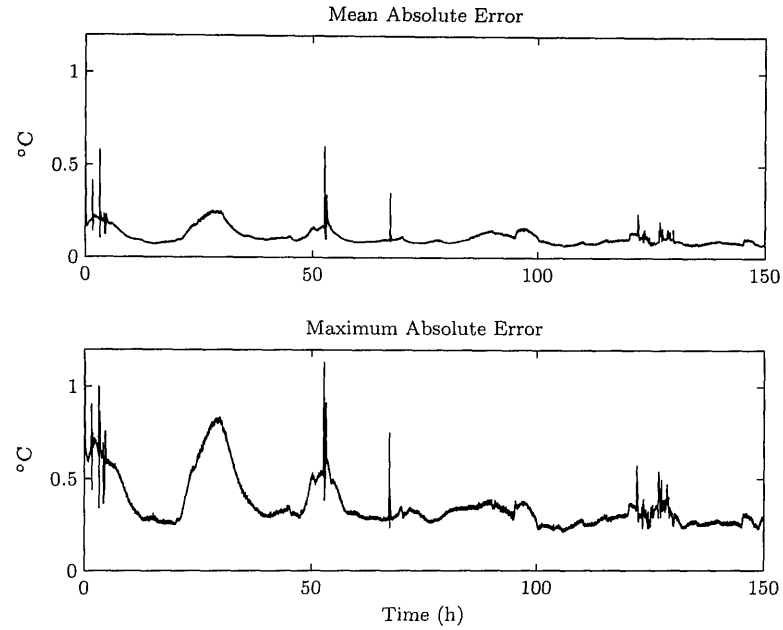

Figure 7.8: Mean (top figure) and maximum (bottom figure) absolute error from the target temperature for all the actuators in the installation.

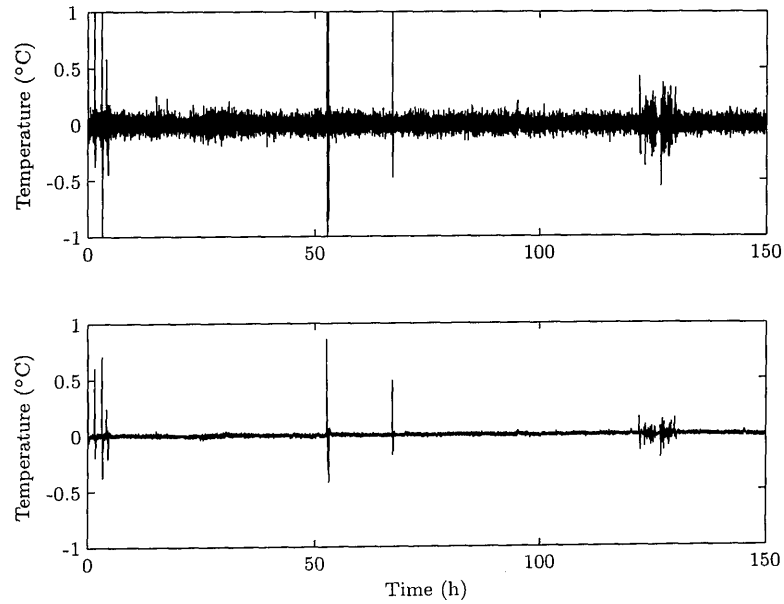

Figure 7.9: The rate–of–change of the inlet temperature for the controller of Wang and Young (2006) (top figure) and the proposed controller that explicitly accounts for the disturbance model (Section 7.2.2) (bottom figure).

the state vector with measurements of the disturbance signal and an alternative that evaluates the disturbance model directly into the prediction equations. In both methods, the state vector consists only of directly measurable variables, thus eliminating the need of a state reconstuctor. Although both solutions are already available in the literature, they are presented to extend the functionality of the NMSS/MPC controller of Wang and Young (2006). In general, the approaches presented here can use the added functionality of the NMSS/MPC structures presented in this thesis in cases were a measured modelled disturbance signal is available.

Furthermore, a simulation case study is considered were the complete estimation and control procedure is presented. The simulation is based on data that were collected from an actual temperature control installation and the environmental temperature is considered as a disturbance signal. The procedure to identify the variables to be controlled is presented and simulations show that the proposed technique results in an acceptable closed loop response. Finally, comparison with the controller of Wang and Young (2006) that does not account for the disturbance model shows that the controller presented in this chapter produces smoother control signals.

Chapter 8

MPC for State Dependent Parameter Models

8.1 Introduction

This chapter considers constraint handling techniques to the non–linear class of State Dependent Parameter (SDP) models. The SDP methodology introduced by Young (1969) (see also Young et al., 2001, and references therein) are state space models whose parameters depend on elements of the state vector. In the NMSS framework, this effectively means that the system parameters are functions of previous measurements of the input or output variables. There are close parallels of SDP systems with Time Variable Parameter (TVP) systems. However, in TVP systems the system parameters vary slowly while the state dependency of the parameters in SDP systems can lead to rapid changes in the system dynamics. This allows for SDP models to describe non–linear processes that can include chaotic systems and systems that have previously been modelled using a bilinear approach (Dunoyer et al., 1997).

Within the SDP literature, some methods have been presented for estimation of such models (Åkesson and Toivonen, 2006; Toivonen, 2003; Toivonen et al., 2007; Young et al., 2001) that consist of a variety of approaches to the identification

problem. Furthermore, the linear–like structure of SDP systems allows for them to be considered at each sampling instant as *frozen* linear, instances of the non–linear system. Following this approach, PIP control methodologies have been studied (Kontoroupis et al., 2003; Taylor, 2005) and applied to practical systems (Stables and Taylor, 2006; Taylor et al., 2006a). Based on such a PIP controlled SDP system, Section 8.2 evaluates the Reference Governor (RG) approach presented in Chapter 3 to deal with constraints in SDP control system design.

Next, Sections 8.3 and 8.4 present two approaches that directly apply MPC to the SDP system. As already stated before, there are close parallels of SDP systems with TVP systems. However, in TVP systems the system parameters vary slowly (see Guo and Rugh, 1995; Hunt and Johansen, 1997; Shamma and Xiong, 1999; Stilwell and Rugh, 2002, for some examples of dealing with stability of TVP systems) while the parameter state dependency in SDP systems can lead to rapid changes in the system dynamics. Since stability results of TVP systems are based on the slow parameter variation, they cannot be used in analysing stability for SDP models. Earlier work of Kouvaritakis et al. (1999) has addressed the issue of stability of MPC controlled non–linear systems by linearising at every sampling instant. Although this approach is eventually based on a piecewise linear model (as in the case of SDP systems), their analysis requires a control trajectory for the initial non–linear system which is not the way the non–linearity is dealt with here.

The piecewise linear nature of SDP systems allows for concepts of linear control theory to be applied (in contrast to the approach of Kouvaritakis et al. (1999) where the linearised model is an approximation of the actual system). For example, Kontoroupis et al. (2003) have considered a *frozen* system, solving the Linear Quadratic (LQ) problem at each sampling instant, while Taylor (2005) parametrically defines the state feedback control law for closed loop pole placement. Since MPC involves solving a Quadratic Program (QP) at every sampling instant, the former approach is adopted in the present chapter.

However, by not taking into account the evolution of the system in the predictions, essentially introduces model mismatch and error in the predictions that can be considerable depending on the degree of non–linearity. In this regard, Sections 8.3.2 and 8.4.2 present alterations to the initial algorithm to take into account the non–linearity. The underlying idea is to use the trail of the control trajectory (i.e. the predicted control moves that were not applied to the system) at the previous sampling instant as an initial condition to estimate an initial *trajectory* for the system and use it to calculate an optimising control sequence. This sequence can again be used to update the system *trajectory* and a new optimal control sequence can be calculated. This procedure can iteratively be used until the optimising control sequence (or the system trajectory) converges. The drawback of this procedure is that it is computationally more demanding than the one that does not take into account the non–linearity, and requires the convergence of the optimising control law and the system output.

The techniques presented in Sections 8.3 and 8.4 are shown to be stable under assumptions very similar to the ones already used in linear MPC. The improvement in the predictions by recursive solution of the QP as presented in Sections 8.3.2 and 8.4.2 along with the stability results of Sections 8.3.3 and 8.4.3 are the major contributions of this chapter.

8.2 The Reference Governor approach

This section presents the RG technique introduced in Chapter 3 for the case where the system is non–linear and described in SDP form. As in Chapter 3 a pre–stabilising PIP controller is required that gives the closed loop system it's desired properties in the absence of constraints. Then, the RG accounts for system constraints by changing the reference signal trajectory. More information on the RG approach to non–linear systems can be found in Bemporad (1998a) with a detailed analysis of it's properties (still with a different parametrisation of the degrees of

freedom as previously discussed in Chapter 3).

The general SDP system can be described by the following difference equation that is clearly in the form of (2.1) with the system parameters changing at each sampling instant:

$$y_k = \sum_{i=1}^{n} -a_i(\chi_{i,k})y_{k-i} + \sum_{i=1}^{m} b_i(\psi_{i,k})u_{k-i} \tag{8.1}$$

where $\chi_{i,k}$ and $\psi_{i,k}$ can be functions of previous values of the input or output. For simplicity in the notation, in the following $a_i(\chi_{i,k})$ and $b_i(\psi_{i,k})$ are referred to as $a_{i,k}$ and $b_{i,k}$ respectively.

The above, can be described by the following piecewise linear state space form:

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{b}_k u_k + \mathbf{d}r_k \tag{8.2a}$$

$$y_{k+1} = \mathbf{c}\mathbf{x}_{k+1} \tag{8.2b}$$

where the state vector is given by (3.2) and $\mathbf{A}_k$ and $\mathbf{b}_k$ are the system matrices that are dependent on some state variables and change at each sampling instant[1]. The system matrix and vectors are the same as in (3.3) but with varying parameters where necessary.

Subsequently, a SDP/PIP controller is designed for the system. Various approaches to this exist, a review of which can be found in Kontoroupis et al. (2003) and a more detailed study of the SDP pole placement technique in Taylor (2005). In any case, a *frozen* system is considered at each sampling instant and the desired control design technique is applied. The state feedback controller in every case takes the form:

$$u_k = -\mathbf{k}_k\mathbf{x}_k \tag{8.3}$$

[1]Depending on the form of (8.1) one of $\mathbf{A}_k$ and $\mathbf{b}_k$ may not be varying, but in the general case described here, both change at every sampling instant.

where the state gain vector $\mathbf{k}_k$ is varying to account for the system non–linearity. After application of the above control law, the varying system (8.2) takes the form of (3.5) with constant system matrices. More specifically, since the $\mathbf{k}_k$ vector changes at each sampling instant to achieve the closed loop performance objectives, the $\tilde{\mathbf{A}} = \mathbf{A}_k - \mathbf{b}_k\mathbf{k}_k$ matrix is now constant and defined by the closed loop pole positions.

The problem formulation within the MPC framework is identical to the one presented in Section 3.2 and is omitted here for brevity. The only difference lies in the varying nature of $\mathbf{k}_k$ that results in $\mathbf{K}_1$ and $\mathbf{K}_2$ of (3.9) to be varying. The new reference signal that is then applied to the system is derived by (3.6) after solving the optimisation problem (3.12) at every sampling instant.

8.3 MPC with an explicit integral–of–error state

Since SDP systems have been extensively studied in the NMSS framework, one straightforward and well studied model description is the one already presented for PIP control in Section 3.1, MPC control in Section 4.2 and SDP/PIP control in Section 8.2. In this regard, the system description with an explicit integral–of–error state presented in Section 8.2 is used to form an MPC controller for SDP systems.

8.3.1 The basic algorithm

As in Chapter 4, the controller can be derived by numerical solution of the optimisation problem (4.12) at every sampling instant. To formulate the prediction equations, the procedure similar to the one described in Section 4.2 is followed. In this regard the future state, the future reference and future input vectors are defined as in (4.7a), (4.9) and (4.13) respectively. Assuming a *frozen* model (i.e. the use of the same model throughout the prediction horizon), the prediction equations

at each sampling instant can be written as:

$$\mathbf{X} = \mathbf{F}_k \mathbf{x}_k + (\Phi_k + \Phi_{1,k})\,\mathbf{U} + \mathbf{H}_{r,k}\mathbf{S} \tag{8.4}$$

in which $\mathbf{F}_k$, $\mathbf{\Phi}_k$, $\mathbf{\Phi}_{1,k}$ and $\mathbf{H}_{r,k}$ are varying versions of $\mathbf{F}$, $\mathbf{\Phi}$, $\mathbf{\Phi}_1$ and $\mathbf{H}_r$ of (4.14) that account for the change of the SDP system at every sampling instant.

In a similar manner to Section 4.2, the optimisation problem can take the matrix form of (4.16).

8.3.2 Improving the predictions

From the above, it is clear that the accuracy of the state predictions depends on the non–linearity of the system and the distance from the set–point since only the present frozen system is considered (this is made clear later with some simulation examples). Although the system still moves to the predicted state at the following sampling instant, the predictions for more than one step ahead into the future are bound to some error due to the change in the system matrices. Especially in cases where long horizons are used, although the control action that is computed at some sampling instant is optimal at that time, it may not even be feasible at the next one even in the absence of disturbances. It is therefore not optimal in the sense that it does not take into account the inherent non–linearity of the system.

In this regard, a methodology is presented where the prediction accuracy is drastically improved (when the proposed algorithm converges). The underlying idea is that starting from an initial control sequence, an estimate of the predictions can be found that can then be used to estimate the evolution of the system. The same procedure can be iteratively evaluated until the optimal control sequence (and hence the predicted system output) converges[2]. The predicted states are in this case much closer (identical in the case of no model mismatch and the absence

[2]As suggested later, the convergence of the proposed algorithm is not guaranteed and the claims made refer to the cases when the algorithm has converged.

of disturbances) to the actual ones. The resulting control law can therefore be considered optimal not only for the *frozen* system, but also for the actual non–linear system.

Defining as $\mathbf{A}_{k+i|k}$ and $\mathbf{b}_{k+i|k}$ the estimates of $\mathbf{A}_{k+i}$ and $\mathbf{b}_{k+i}$ from measurements at the k-th sampling instant and application of a feasible control trajectory $\{u_{k|k}, u_{k+1|k}, \ldots, u_{k+N_c-1|k}\}$, the matrices in the prediction equation (8.4) can be written as:

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{Z}_k^{1,0} \\ \mathbf{Z}_k^{2,0} \\ \vdots \\ \mathbf{Z}_k^{N_p,0} \end{bmatrix}; \qquad \mathbf{H}_{r,k} = \begin{bmatrix} \mathbf{d} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{Z}_k^{2,1}\mathbf{d} & \mathbf{d} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{Z}_k^{3,1}\mathbf{d} & \mathbf{Z}_k^{3,2}\mathbf{d} & \mathbf{d} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_k^{N_p,1}\mathbf{d} & \mathbf{Z}_k^{N_p,2}\mathbf{d} & \mathbf{Z}_k^{N_p,3}\mathbf{d} & \cdots & \mathbf{d} \end{bmatrix} \tag{8.5a}$$

$$\Phi_k = \begin{bmatrix} \mathbf{b}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{Z}_k^{2,1}\mathbf{b}_k & \mathbf{b}_{k+1|k} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_k^{N_c,1}\mathbf{b}_k & \mathbf{Z}_k^{N_c,2}\mathbf{b}_{k+1|k} & \cdots & \mathbf{b}_{k+1|k} \\ \mathbf{Z}_k^{N_c+1,1}\mathbf{b}_k & \mathbf{Z}_k^{N_c+1,2}\mathbf{b}_{k+1|k} & \cdots & \mathbf{Z}_k^{N_c+1,N_c}\mathbf{b}_{k+1|k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Z}_k^{N_p,1}\mathbf{b}_k & \mathbf{Z}_k^{N_p,2}\mathbf{b}_{k+1|k} & \cdots & \mathbf{Z}_k^{N_p,N_c}\mathbf{b}_{N_c-1|k} \end{bmatrix}; \; \Phi_{1,k} = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & \mathbf{W}_k^{N_c+1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \mathbf{W}_k^{N_p} \end{bmatrix} \tag{8.5b}$$

where

$$\prod_{l=i}^{j} \mathbf{A}_{l|k} = \mathbf{A}_{j|k}\mathbf{A}_{j-1|k}\cdots\mathbf{A}_{i|k}$$

$$\mathbf{Z}_k^{i,j} = \prod_{l=j+1}^{i} \mathbf{A}_{k+l-1|k}$$

$$\mathbf{W}_k^i = \mathbf{b}_{k+i-1|k} + \sum_{l=1}^{i-N_c-1} \mathbf{Z}_k^{i-N_c-1,N_c+l}\mathbf{b}_{k+N_c+l-1|k}$$

It is straightforward to identify an iterative procedure to calculate $\mathbf{Z}_k^{i,j}$ and $\mathbf{W}_k^i$. Starting from $\mathbf{Z}_k^{1,0} = A_k$ and $\mathbf{W}_k^{Nc+1} = \mathbf{b}_{k+N_c|k}$ the following iterations can be followed:

$$\mathbf{Z}_k^{i+1,j} = \mathbf{A}_{k+i|k}\mathbf{Z}_k^{i,j}$$

$$\mathbf{Z}_k^{i,j+1} = \mathbf{Z}_k^{i,j}\mathbf{A}_{k+j-1|k}^{-1}$$

$$\mathbf{W}_k^{i+1} = \mathbf{b}_{k+i|k} + \mathbf{A}_{k+i+1|k}\mathbf{W}_k^i$$

The control signal is again derived by numerical solution of the optimisation problem (4.12) at every sampling instant, but the matrices used in the prediction equation (8.4) are the ones defined by (8.5). Furthermore, the procedure is repeated until the future control signal trajectory converges and then the first one is applied to the plant, while the rest is discarded according to the receding horizon control strategy. To improve the speed of the computations, the trail of the control signal can be used as a starting point for the search of the new optimised control sequence.

The approach is summarised in the next Algorithm.

Algorithm 8.3.1.

1. *Solve the optimisation problem* (4.12) *using the current measurement of the state to calculate a control sequence* $\{u_k, u_{k+1}, \ldots, u_{k+N_c-1}\}$.

2. *Use the calculated control sequence* $\{u_k, u_{k+1}, \ldots, u_{k+N_c-1}\}$ *to obtain an estimate of the evolution of the system matrices* $\{\mathbf{A}_k, \mathbf{A}_{k+1}, \ldots, \mathbf{A}_{k+N_p}\}$ *and* $\{\mathbf{b}_k, \mathbf{b}_{k+1}, \ldots, \mathbf{b}_{k+N_p}\}$ *where* $u_{k+i} = u_{k+N_c-1}$ *for* $i = N_c, \ldots, N_p - 1$.

3. *Solve the optimisation problem* (4.12) *using the matrices defined in* (8.5) *and calculate a new control sequence.*

4. *If the new control sequence calculated in the previous step is the same as the one calculated before, exit the algorithm and apply the first element to the system. Else GOTO step 2.*

8.3.3 Stability analysis

This section follows the procedure used in MPC of linear systems (Appendix C) to obtain stability results for the present MPC control scheme of non–linear SDP models. In this regard, the following assumptions are made:

Assumption 8.3.1. *For each sampling instant k, there exists a solution* $\mathbf{U}$ *to the optimisation problem* (4.12).

Assumption 8.3.2. *For each sampling instant k+1, the control sequence calculated at sampling instance k is feasible when appended with it's last value (i.e.* $\mathbf{U}(k+1) = \{\mathbf{U}(k)_2^{N_c-1}, u_{k+N_c-1}\}$*).*

where $\mathbf{U}_i^j$ refers to the ith till jth elements of $\mathbf{U}$.

Assumption 8.3.3. *The resulting control law leads to future state that satisfies:*

$$\mathbf{x}_{k+N_p+1}^T \mathbf{Q} \mathbf{x}_{k+N_p+1} + r u_{k+N_c-1}^2 - \mathbf{x}_{k+1}^T \mathbf{Q} \mathbf{x}_{k+1} - r u_k^2 \leq 0$$

The following proposition presents a stability analysis for the SDP/MPC control scheme presented above when the predictions are calculated using Algorithm 8.3.1. It should be noted here that convergence of Algorithm 8.3.1 is necessary for the following proposition to hold.

Proposition 8.3.1. *Subject to Assumptions 8.3.1, 8.3.2 and 8.3.3 the closed loop model predictive control system based on a system with an explicit integral–of–error state is asymptotically stable.*

Proof. The analysis is based on Lyapounov stability theory and is very similar to the stability proof of Theorem C.2.1. The differences are highlighted below.

In this case, the sequence

$$\mathbf{u}_{|k+1\rightarrow}\left\{u^*_{k+1|k}, u^*_{k+2|k}, \ldots, u^*_{k+N_c-1|k}, u^*_{k+N_c-1|k}\right\} \tag{8.6}$$

is considered as feasible at the $k+1$ sampling instance (Assumption 8.3.2). Based on (8.6), equation (C.3) becomes:

$$\begin{aligned}
\Delta V &= V(\mathbf{x}_{k+1}, \mathbf{u}^*_{|k+1\rightarrow}) - V(\mathbf{x}_k, \mathbf{u}^*_{|k\rightarrow}) \\
&\leq \tilde{V}(\mathbf{x}_{k+1}, \mathbf{u}_{|k+1\rightarrow}) - V(\mathbf{x}_k, \mathbf{u}^*_{|k\rightarrow}) \\
&= \mathbf{x}^T_{k+N_p+1|k+1}\mathbf{Q}\mathbf{x}_{k+N_p+1|k+1} + ru^{*2}_{k+N_c-1|k} - \mathbf{x}^T_{k+1|k}\mathbf{Q}\mathbf{x}_{k+1|k} - ru^{*2}_{k|k}
\end{aligned}$$

From Assumption 8.3.3 it follows:

$$V(\mathbf{x}_{k+1}, \mathbf{u}^*_{|k+1\rightarrow}) - V(\mathbf{x}_k, \mathbf{u}^*_{|k\rightarrow})) \leq 0$$

that, along with the fact the V is positive definite, completes the proposition. □

Remark 8.3.1. *From Proposition 8.3.1 it follows the the proposed controller is asymptotically stabilising. However, Assumption 8.3.3 is not very straightforward to guarantee, especially because of the presence of the integral–of–error state. In order to avoid such a requirement for asymptotic stability to be proved, the next section presents an alternative formulation of the problem that uses assumptions identical to the case of MPC of linear systems (although convergence of Algorithm 8.3.1 is still assumed).*

8.4 MPC Formulation using an increment in the control action

This section presents a formulation of the SDP/MPC problem based on a system description that parametrises the control signal using it's increments. This problem formulation allows for asymptotic stability to be proved based on assumptions identical to the case of MPC of linear systems. The analysis that follows still assumes convergence of the predictions in the same manner as in Algorithm 8.3.1.

8.4.1 The basic algorithm

To avoid Assumption 8.3.3 at the stability proposition of the SDP/MPC scheme, the system could be described as in Wang and Young (2006), based on differences in the output and the control action. However, a system of that form is in general not possible to be derived from (8.1). Motivated by Maciejowski (2002, Section 2.6), the control signal is parametrised in terms of the control increment as follows:

$$
\begin{aligned}
u_k &= u_{k-1} + \Delta u_k \\
u_{k+1} &= u_{k-1} + \Delta u_k + \Delta u_{k+1} \\
&\vdots \\
u_{k+i} &= u_{k-1} + \sum_{j=0}^{i} \Delta u_{k+j}
\end{aligned}
$$

Using the above parametrisation for the control signal, the state prediction equations (8.4) can be described in terms of the control increment analytically as follows

(assuming the *frozen* system $\{\mathbf{A}_k,\mathbf{b}_k\}$):

$$\mathbf{x}_{k+1|k} = \mathbf{A}_k\mathbf{x}_k + \mathbf{b}_k\left(u_{k-1} + \Delta u_{k|k}\right) + \mathbf{d}r_{k+1} \tag{8.7a}$$

$$\mathbf{x}_{k+2|k} = \mathbf{A}_k\mathbf{x}_{k+1|k} + \mathbf{b}_k\left(u_{k-1} + \Delta u_{k|k} + \Delta u_{k+1|k}\right) + \mathbf{d}r_{k+2} \tag{8.7b}$$

$$\vdots$$

$$\mathbf{x}_{k+N_p|k} = \mathbf{A}_k\mathbf{x}_{k+N_p-1|k} + \mathbf{b}_k\left(u_{k-1} + \Delta u_{k|k} + \ldots + \Delta u_{k+N_c-1|k}\right) + \mathbf{d}r_{k+N_p} \tag{8.7c}$$

By recursive application of (8.7) (i.e. substituting (8.7a) to (8.7b) and so on), the general form of an i–step ahead prediction of the state vector can be presented as:

$$\mathbf{x}_{k+i|k} = \mathbf{A}_k^i\mathbf{x}_k + \sum_{j=0}^{i-1}\mathbf{A}_k^j\mathbf{b}_k u_{k-1} + \sum_{l=0}^{i-1}\sum_{j=0}^{i-1-l}\mathbf{A}_k^j\mathbf{b}_k\Delta u_{k+l|k} + \sum_{l=0}^{i-1}\sum_{j=0}^{i-1-l}\mathbf{A}_k^j\mathbf{b}_k r_{k+l+1}$$

in which $\Delta u_{k+i|k}$ is the predicted increment in the control signal and $\Delta u_{k+i|k} = 0$ for $i \geq N_c$.

The controller can subsequently be derived as in Wang and Young (2006) by numerical solution of the optimisation problem defined by the cost function (6.2) and the constraints (4.12b). Defining the future state, the future output and future control increment vectors as in (4.7) and the future reference trajectory vector as in (4.9), the prediction equations can be summarised in:

$$\mathbf{X} = \mathbf{F}_k\mathbf{x}_k + \mathbf{\Phi}_{2,k}u_{k-1} + \mathbf{\Phi}_{3,k}\Delta\mathbf{U} + \mathbf{H}_k\mathbf{S} \tag{8.8a}$$

$$\mathbf{Y} = \bar{\mathbf{C}}\mathbf{X} \tag{8.8b}$$

in which

$$\mathbf{\Phi}_{2,k} = \begin{bmatrix} \mathbf{b}_k \\ \mathbf{A}_k\mathbf{b}_k + \mathbf{b}_k \\ \vdots \\ \sum_{i=0}^{N_c-1} \mathbf{A}_k^i\mathbf{b}_k \\ \sum_{i=0}^{N_c} \mathbf{A}_k^i\mathbf{b}_k \\ \vdots \\ \sum_{i=0}^{N_p-1} \mathbf{A}_k^i\mathbf{b}_k \end{bmatrix}; \mathbf{\Phi}_{3,k} = \begin{bmatrix} \mathbf{b}_k & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_k\mathbf{b}_k + \mathbf{b}_k & \mathbf{b}_k & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{N_c-1} \mathbf{A}_k^i\mathbf{b}_k & \sum_{i=0}^{N_c-2} \mathbf{A}_k^i\mathbf{b}_k & \cdots & \mathbf{b}_k \\ \sum_{i=0}^{N_c} \mathbf{A}_k^i\mathbf{b}_k & \sum_{i=0}^{N_c-1} \mathbf{A}_k^i\mathbf{b}_k & \cdots & \mathbf{A}_k\mathbf{b}_k \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=0}^{N_p-1} \mathbf{A}_k^i\mathbf{b}_k & \sum_{i=0}^{N_p-2} \mathbf{A}_k^i\mathbf{b}_k & \cdots & \sum_{i=0}^{N_p-N_c} \mathbf{A}_k^i\mathbf{b}_k \end{bmatrix}$$

And $\bar{\mathbf{C}}$ is a block diagonal matrix with the vector $\mathbf{c}$ on it's diagonal. The relationship between the future control increment and the future control signal vectors can subsequently be defined as:

$$\mathbf{U} = \mathbf{C}_3 u_{k-1} + \mathbf{C}_4 \mathbf{\Delta U}$$

where:

$$\mathbf{C}_3 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}; \qquad \mathbf{C}_4 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

The optimisation problem can subsequently be written in matrix form as:

$$\begin{cases} \min_{\Delta \mathbf{U}} J = (\mathbf{S} - \mathbf{Y})^T \bar{\mathbf{Q}} (\mathbf{S} - \mathbf{Y}) + \Delta \mathbf{U}^T \bar{\mathbf{R}} \Delta \mathbf{U} \\ s.t. \ \tilde{\mathbf{M}} \Delta \mathbf{U} \leq \tilde{\mathbf{N}} \end{cases} \tag{8.9}$$

where the matrix $\tilde{\mathbf{M}}$ and vector $\tilde{\mathbf{N}}$ define the constraints on the input, it's incre-

ments and the output as follows:

$$\tilde{\mathbf{M}} = \begin{bmatrix} -\mathbf{C}_4 \\ \mathbf{C}_4 \\ -\mathbf{I}_{N_c} \\ \mathbf{I}_{N_c} \\ -\bar{\mathbf{C}}\boldsymbol{\Phi}_{3,k} \\ \bar{\mathbf{C}}\boldsymbol{\Phi}_{3,k} \end{bmatrix}; \qquad \tilde{\mathbf{N}} = \begin{bmatrix} -\underline{\mathbf{U}} + \mathbf{C}_3 u_{k-1} \\ \overline{\mathbf{U}} - \mathbf{C}_3 u_{k-1} \\ -\underline{\boldsymbol{\Delta}\mathbf{U}} \\ \overline{\boldsymbol{\Delta}\mathbf{U}} \\ -\underline{\mathbf{Y}} + \bar{\mathbf{C}}\mathbf{F}_k\mathbf{x}_k + \bar{\mathbf{C}}\boldsymbol{\Phi}_{2,k}u_{k-1} + \bar{\mathbf{C}}\mathbf{H}_r\mathbf{S} \\ \overline{\mathbf{Y}} - \bar{\mathbf{C}}\mathbf{F}_k\mathbf{x}_k - \bar{\mathbf{C}}\boldsymbol{\Phi}_{2,k}u_{k-1} - \bar{\mathbf{C}}\mathbf{H}_r\mathbf{S} \end{bmatrix}$$

8.4.2 Improving the predictions

In the same principle as in Section 8.3.2, the predictions can be improved by taking into account the evolution of the system. In this regard, the predictions are still based on (8.8) with the matrices $\mathbf{F}_k$ and $\mathbf{H}_{r,k}$ as in (8.5a) while $\boldsymbol{\Phi}_{2,k}$ and $\boldsymbol{\Phi}_{3,k}$ are defined as:

$$\boldsymbol{\Phi}_{2,k} = \begin{bmatrix} \tilde{\mathbf{W}}_k^{1,1} \\ \tilde{\mathbf{W}}_k^{2,1} \\ \vdots \\ \tilde{\mathbf{W}}_k^{N_p,1} \end{bmatrix}; \qquad \boldsymbol{\Phi}_{3,k} = \begin{bmatrix} \tilde{\mathbf{W}}_k^{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{\mathbf{W}}_k^{2,1} & \tilde{\mathbf{W}}_k^{2,2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{W}}_k^{N_p,1} & \tilde{\mathbf{W}}_k^{N_p,2} & \cdots & \tilde{\mathbf{W}}_k^{N_p,N_c} \end{bmatrix} \tag{8.10}$$

in which

$$\tilde{\mathbf{W}}_k^{i,j} = \mathbf{b}_{k+i-1|k} + \sum_{l=j}^{i-1} \mathbf{Z}_k^{i,j} \mathbf{b}_{k+l-1|k}$$

An iterative procedure to calculate $\tilde{\mathbf{W}}_k^{i,j}$ can subsequently be defined by the following equation:

$$\begin{aligned} \tilde{\mathbf{W}}_k^{i+1,j} &= \mathbf{A}_{k+i|k} \tilde{\mathbf{W}}_k^{i,j} + \mathbf{b}_{k+i|k} \\ \tilde{\mathbf{W}}_k^{i,j+1} &= \tilde{\mathbf{W}}_k^{i,j} - \mathbf{Z}^{i,j} \mathbf{b}_{k+j-1|k} \end{aligned}$$

where $\tilde{\mathbf{W}}_k^{1,1} = \mathbf{b}_k$. The control action is again derived by numerical optimisation of (8.9) using the matrices of (8.5a) and (8.10) in the prediction equations (8.8). A summary of the above can take the form of Algorithm 8.3.1[3] and is omitted here for brevity.

8.4.3 Stability analysis

Stability of the presented algorithm follows in a very similar way as in Section 6.1.1. However, for completeness and to demonstrate that the assumptions take a similar form as in the linear case for this class of non–linear systems when the improved predictions of Section 8.4.2 are used (and Algorithm 8.3.1 has converged), the analysis is repeated.

Proposition 8.4.1. *Subject to Assumptions C.2.1 and C.2.2 and convergence of Algorithm 8.3.1 the closed loop model predictive control SDP system based on a model that is described with increments in the control signal is asymptotically stable.*

Proof. The cost function (6.2) is chosen as a candidate Lyapounov function. From this point on, the analysis is the same as that of Theorem 6.1.1. However, there is still the issue of feasibility sample $k+1$ of the control increment at sample k appended with 0. This is a direct result of the convergence of Algorithm 8.3.1, which completes the analysis. □

Remark 8.4.1. *Sections 8.3.2 and 8.4.2 presented techniques to improve the prediction equations that both require knowledge of the evolution of the prediction matrices. As results from Algorithm 8.3.1, this requires the loop that comprises of Steps 2–4 to have converged to a solution for the future control signal (or control increment). This part of the algorithm has no guarantee of convergence and can be restrictive in practical applications. It still needs to be considered regarding it's*

[3]Since the decision variables are the control increment and the control action itself, Algorithm 8.3.1 can be adapted for this case by changing u to Δu.

speed (number of iterations needed) and it's convergence properties. However, for the analysis of the present chapter it is regarded that the algorithm has converged to a solution.

8.5 Simulation Examples

This section presents two simulation examples for the constraint handling approaches within the SDP framework that were described in the previous sections. Section 8.5.1 briefly presents the design process of an SDP/PIP controller for pole placement that does not account for system constraints and it is then shown that the introduction of a RG maintains the designed properties of the closed loop system in the presence of constraints. Next, Section 8.5.2 evaluates the SDP/MPC techniques of Sections 8.3 and 8.4 for a non–linear SDP system and their differences are highlighted.

8.5.1 SDP/PIP control and Reference Management

Consider the bilinear system described by the difference equation:

$$y_k = 0.7y_{k-1} + y_{k-2}u_{k-1} \tag{8.11}$$

that is of the general form (8.1) with $a_{1,k} = -0.7$ and $b_{1,k} = y_{k-2}$. Choosing the state vector to be:

$$\mathbf{x}_k = \begin{bmatrix} y_k \\ z_k \end{bmatrix} \tag{8.12}$$

the system (8.11) can be described by the state space form (8.2) where the state matrices and vectors are given by:

$$\mathbf{A} = \begin{bmatrix} 0.7 & 0 \\ -0.7 & 1 \end{bmatrix}; \qquad \mathbf{b}_k = \begin{bmatrix} y_{k-1} \\ -y_{k-1} \end{bmatrix}; \qquad \mathbf{d} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \qquad \mathbf{c} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

As discussed in Section 8.2, for SDP/PIP control, the control action has the state feedback form of (8.3). In this example, the pole placement approach described in Kontoroupis et al. (2003) and Taylor (2005) is considered and is briefly described in the following. The design objective is for the closed loop system to have a response of one with a characteristic polynomial of the form:

$$D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} \tag{8.13}$$

Analytical calculation of the closed loop characteristic polynomial using the control action (8.3) and comparison to (8.13) results to the following feedback gain vector:

$$\mathbf{k}_k = \begin{bmatrix} \frac{0.7-d_2}{y_{k-1}} & -\frac{1+d_1+d_2}{y_{k-1}} \end{bmatrix}$$

In order to highlight the differences between a closed loop system with and without a RG, a deadbeat response (i.e. $d_1 = d_2 = 0$) is chosen as the desired response for the closed loop system. In the absence of constraints the close loop response to a step change in the reference signal from 1 unit to 2 units is depicted in Figure 8.1 (with initial conditions $(y_0, u_0, z_0) = (1, 0.3, 0)$).

Next, an input constraint of 0.5 units (i.e. $|u_k| \leq 0.5$) is imposed to the controlled input. The system response for the same reference step is shown in Figure 8.2 for both the system with (a prediction horizon of $N_p = 3$ is used in this example) and without reference management. It should be noted that to avoid integral windup the incremental form of the SDP/PIP controller is evaluated as described by (3.13). From Figure 8.2 it could be argued that in the absence of the RG the closed loop system has a faster rise time, makes more use of the available control signal and therefore resembles more to the deadbeat response. However, it presents an overshoot that is clearly not desired when a system is designed to have a deadbeat response. In the presence of the RG at every sampling instant, the system has a deadbeat response to the reference presented to it.

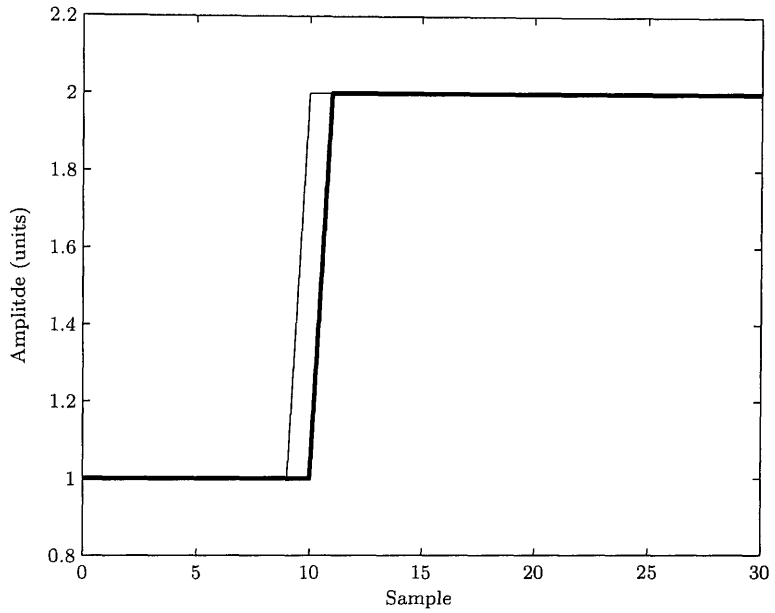

Figure 8.1: Response of the closed loop SDP/PIP control system without constraints. Reference (thin line) and system output (thick line).

8.5.2 SDP/MPC control

Consider the following non–linear SDP system described by the difference equation:

$$y_k = 0.7y_{k-1} + u_{k-2}u_{k-1}$$

that is of the form (8.1) with $a_{1,k} = -0.7$ and $b_{1,k} = u_{k-2}$. The above system can be described by the state space form (8.2), where the state vector is as in (8.12) and the system matrices are given by:

$$\mathbf{A} = \begin{bmatrix} 0.7 & 0 \\ -0.7 & 1 \end{bmatrix}; \qquad \mathbf{b}_k = \begin{bmatrix} u_{k-1} \\ -u_{k-1} \end{bmatrix}; \qquad \mathbf{d} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \qquad \mathbf{c} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

For the MPC controller presented in Section 8.3, the weightings are chosen to be $\mathbf{Q} = \left[\begin{smallmatrix} 3 & 0 \\ 0 & 0.1 \end{smallmatrix}\right]$ and $R = 1$; the prediction and control horizons are set to $N_p = 10$ and $N_c = 8$ respectively and the initial condition of the system is $(y_0, u_0, u_{-1}, z_0) = (1, 0.547, 0.547, 0)$. Figure 8.3 presents the closed loop response of the MPC system

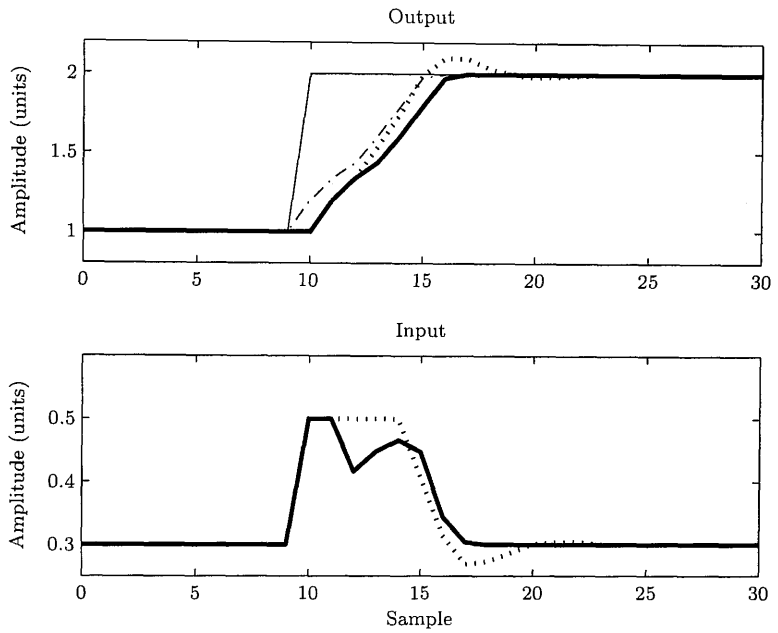

Figure 8.2: Response of the SDP/PIP scheme with RG (thick solid line) and without RG (dotted line) in the presence of constraints. The actual reference signal is shown (thin solid line) and the one that is given to the system (thin dash–dotted line) in the case where the RG is used.

to a reference step from 1 unit to 2 units. Furthermore, the control signal is constrained to have amplitude less than 1 unit (i.e. $|u_k| \leq 1$) and the rate–of–change to have amplitude less than 0.2 units/sample (i.e. $|\Delta u_k| \leq 0.2$). It should be noted that the Algorithm 8.3.1 that makes use of the improved predictions of Section 8.3.2 is used for this simulation.

It is clear from Figure 8.3 that a well–behaved stable closed–loop system results from the application of the proposed control system. However, as already mentioned in Remark 8.3.1, the Assumption 8.3.3 is not straightforward to satisfy. To visualise this, Figure 8.4 presents the value of the cost function during the previous simulation. It is evident that the value of the considered Lyapounov function is not monotonically decreasing despite the resulting stable closed loop system. This is the result of Assumption 8.3.3 not being satisfied as already stated.

To avoid this, the parametrisation of Section 8.4 is adopted and the result is

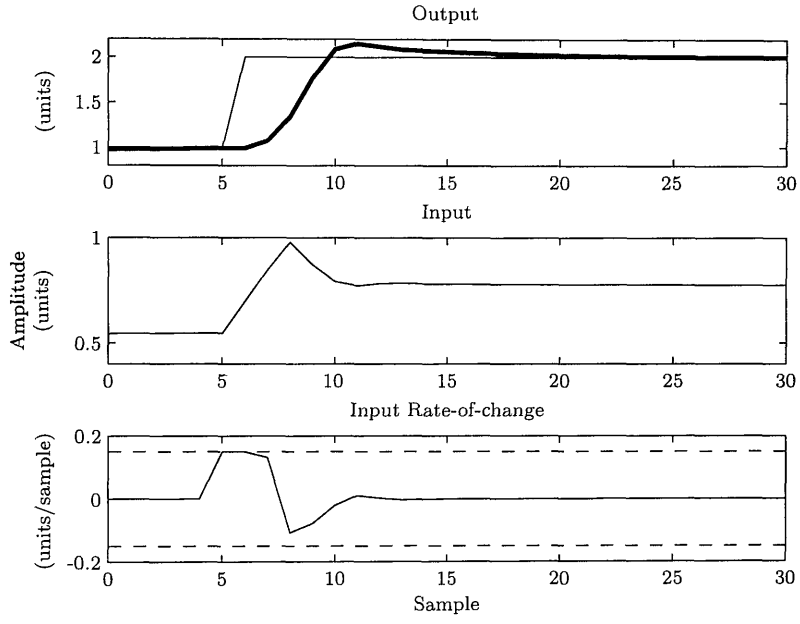

Figure 8.3: Closed loop response for the SDP/MPC control scheme based on a system description with an explicit integral–of–error state in the presence of constraints. The constraint levels are shown as dashed lines.

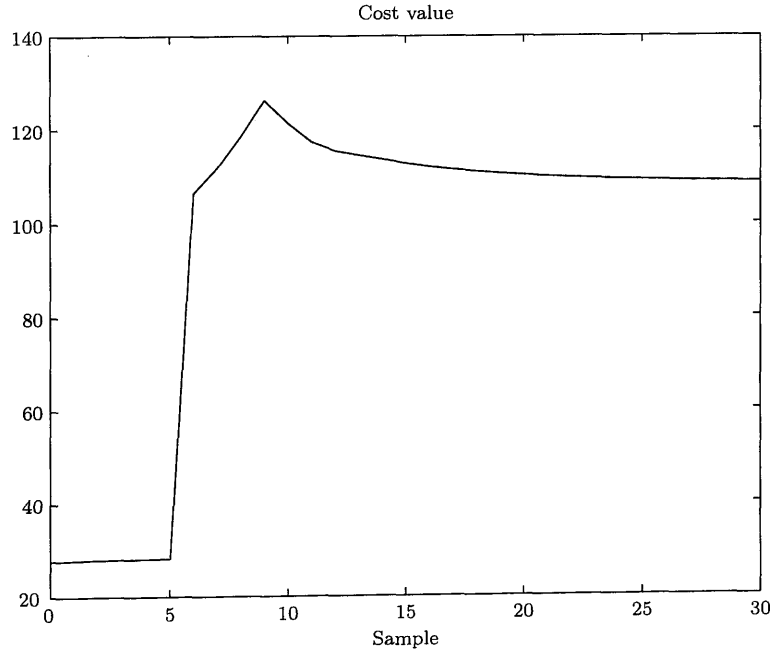

Figure 8.4: The value of the cost function for the SDP/MPC controller presented in Section 8.3 with the improved predictions of Section 8.3.2.

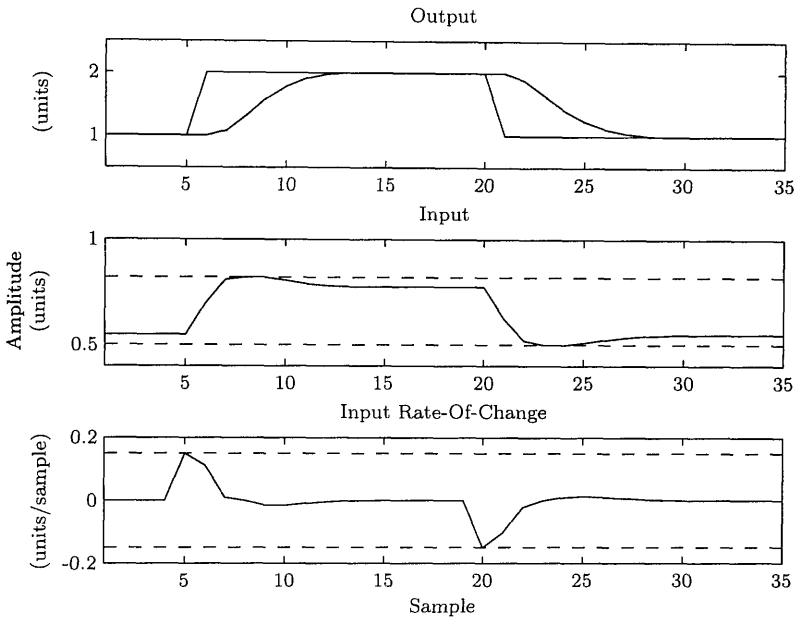

Figure 8.5: Closed loop simulation for the SDP/MPC controller with an increment in the control action *without* the improved predictions of Section 8.4.2. The constraint limits are shown as dashed lines.

presented below. In this case, the constraints are $|u_k| \leq 0.82$ units and $|\Delta u_k| \leq 0.15$ units/sample, while the weightings are chosen to be $Q = 0.1$ and $R = 1$ and the prediction and control horizons are set to $N_p = 10$ and $N_c = 8$ respectively. The closed loop response for the MPC controller *without* the correction of Section 8.4.2 is presented in Figure 8.5. As already mentioned, the predictions at each sampling instant do not follow the actual output of the system. This is depicted in the upper subplot of Figure 8.6 where the *trail* of the predictions is shown for every sampling instant. It can be observed that there is a considerable difference between the predictions of two consecutive sampling instants, even in this case where no disturbances are present. This is due to the evolution of the system (i.e. the variation of the $\mathbf{b}_k$ vector) within the prediction horizon that is not taken into account in the prediction equations in this case.

The improved predictions of Section 8.4.2 are subsequently used (as described by Algorithm 8.3.1) and the system is simulated again. The closed loop response is

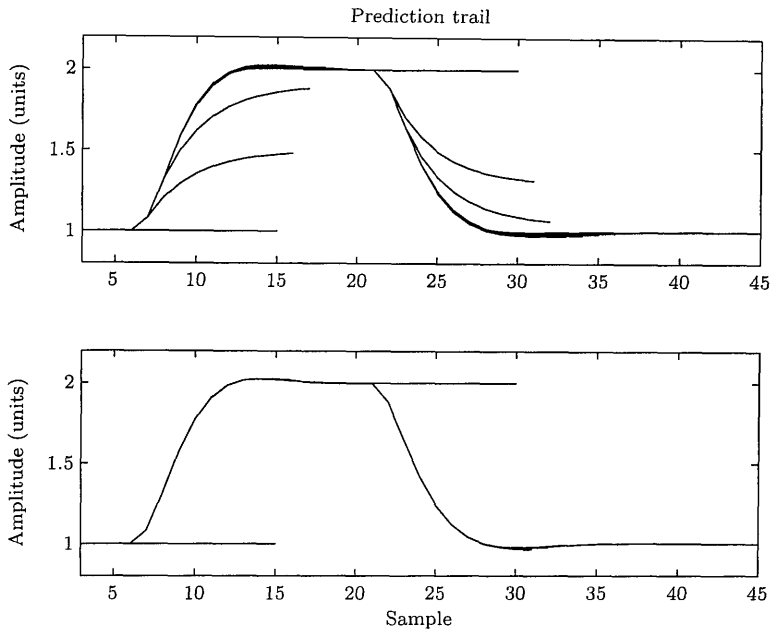

Figure 8.6: Output prediction trail for the SDP/MPC controllers without (top figure) and with (bottom figure) the improvement in the predictions introduced in Section 8.4.2.

very similar to the one depicted in Figure 8.5 and is therefore omitted for brevity. However, as shown at the bottom subplot of Figure 8.6 the predictions are practically indistinguishable from one another visually verifying the fact that Algorithm 8.3.1 has converged. Furthermore, the value of the cost function (that is used as a Lyapounov function in the stability analysis) is presented in Figure 8.7 where it is evident that is monotonically decreasing, resulting in an asymptotically stable closed loop system.

8.6 Concluding remarks

This chapter presented two approaches to constraint handling for non–linear systems described within the SDP framework. The first is a direct application of the RG that was initially presented in Chapter 3 and was adopted here to deal with the varying nature of the SDP models. In this case, a stabilising controller

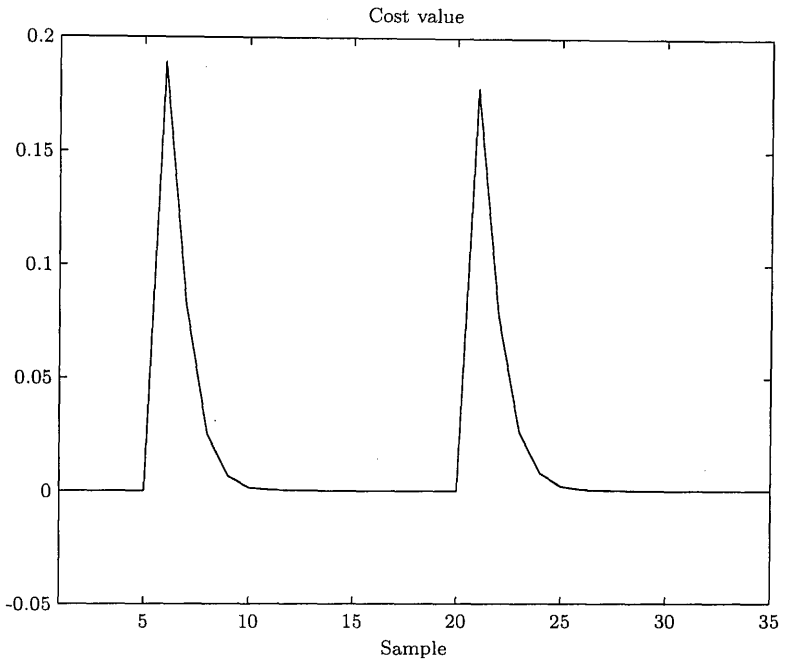

Figure 8.7: The value of the cost function for the SDP/MPC controller with an increment in the control action presented in Section 8.4 with the improved predictions of Section 8.4.2.

(that also gives the closed loop system it's desired properties in the absence of constraints) is necessary and the RG is a high level predictive controller that deals with system constraints. It was shown that the introduction of the RG results in a closed loop system response closer to the desired one (i.e. the one with the stabilising controller in the absence of constraints).

The second approach is a direct application of MPC. In this regard, two different control structures were presented, namely, the form presented in Chapter 4 (adapted to account for the varying nature of SDP systems) and an alternative one that the control signal is parametrised in terms of it's increment. For both, a technique that improves the prediction of the system evolution was presented that makes use of the evolution of the system parameters (that depends on the optimising control sequence). Based on the improved predictions and assuming convergence of the predicted system evolution, conditions for stability of the closed

loop system were presented, that along with a control technique that inherently accounts for system constraints in the design phase in the SDP framework, is the main contribution of this chapter.

Finally, simulation examples were presented to support the claims of the present chapter and highlight the differences between the proposed structures. Furthermore, it is suggested that the structure with the control increment as a decision variable poses more advantages when considering MPC control of SDP systems.

Chapter 9

Conclusion

This thesis has developed a methodology for dealing with constraints in the Non–Minimal State Space (NMSS). Model Predictive Control (MPC) techniques were used and combined with research into NMSS control models, particularly relating to earlier research carried out at Lancaster University into Proportional–Integral–Plus (PIP) control. A summary of the methodological results is presented in Section 9.1, while Section 9.2 gives directions for future research arising from the results in this thesis.

9.1 Summary of results

This section summarises the results of the thesis, which can be broadly divided into four different areas, namely: *supervisory control*; *structural aspects*; *tuning*; and control of *non–linear systems*.

9.1.1 Constraint handling with supervisory control

Constraint handling was the main focus of the present work. Since there has been no attempt for an inherent handling of constraints using NMSS methods before, the first approach focused on a conventional supervisory MPC control scheme, namely the Reference Governor (RG). In this approach, a *high level* controller

that accounts for constraints is utilised on top of an already controlled system. The RG predicts the system evolution and alters the reference signal to avoid constraint violation.

In this thesis, the approach was applied to both linear (Chapter 3) and non–linear (Chapter 8) NMSS systems. In both cases it was shown that the system with the RG results in a closed–loop response that is closer to the desired response (i.e. the response for which the initial controller was designed for) in the presence of constraints. Therefore, it is suggested that in cases where a controller is already available, the addition of a RG can result in constraint satisfaction in an optimal manner (related to the cost function of the RG), whilst still preserving the properties of the *low level* controller when operating away from the constraints.

9.1.2 The importance of structure

The effect of the structure of the NMSS/MPC controller was considered at length in this thesis, following up the initial research by Wang and Young (2006). In Chapter 4 a controller based on a different NMSS system description to the one of Wang and Young (2006) was introduced. This 'integral of error' NMSS/MPC structure allows for the use of tuning techniques already developed for PIP control. Furthermore, it was shown (Chapter 5) that the new structure offers more flexibility and design freedom (resulting from the introduction of the integral–of–error state) when dealing with demanding performance objectives.

A third NMSS/MPC structure, again motivated by PIP control, was subsequently introduced, namely the Forward Path NMSS/MPC controller (Chapter 6). In this case an internal model is used to form part of the state vector (the elements related to the past system outputs). The new approach contrasts with internal model MPC control methods already presented in the literature that estimate the whole state vector. A simulation study comparing Forward Path and the 'Feedback' (Wang and Young, 2006) NMSS/MPC structures yields similar conclusions

to those previously obtained for the equivalent forms of PIP control. More specifically, the Forward Path structure is shown to be more robust when it comes to uncertainty in the numerator parameters, while uncertainty in the denominator results in the closed loop poles to be closer to the desired ones. However, the research also shows that the Feedback structure is most appropriate when dealing with unstable or marginally stable plants, since this form is more likely to yield stable responses. In general, the results for the Forward Path MPC controller mirror those for the Forward Path PIP controller.

Finally, additional NMSS/MPC control structures were considered when a model of a measured disturbance is available. Two different problem formulations were considered, i.e. one that extends the state vector and one that directly accounts for the disturbance measurements in the prediction equations. It was concluded that they both result in similar closed loop performance. The method that directly accounts for disturbance measurements was subsequently considered in a case study relating to a temperature control installation. Here it was shown in simulation to yield improved performance when compared to the NMSS/MPC controller of Wang and Young (2006) (which does not account for the disturbances).

Overall, the thesis concludes that the controller structure and problem formulation affect the performance of NMSS/MPC controllers. Care should be taken when choosing the appropriate control approach since they present different weaknesses and strengths. The most appropriate control structure should be chosen on a case by case basis, depending on the particular problem considered.

9.1.3 Tuning of MPC controllers

One of the contributions of this thesis is the development of an optimal tuning technique for MPC controllers based on goal attainment. Although a similar approach has previously been presented for PID and PIP control system design, Chapter 5 develops the approach for MPC systems. The methodology is shown

to allow for trade–off between various objectives while the controller still accounts for system constraints. Since the tuning is performed off–line, the introduction of an optimisation problem (and therefore an increase in complexity) does not affect the practical applicability of the resulting controller. As mentioned above, it is further demonstrated that the NMSS/MPC control structure presented in Chapter 4 provides the designer with more tuning freedom (than the approach of Wang and Young (2006)) when using the proposed method.

9.1.4 NMSS/MPC control of Non–linear systems

Finally, the thesis considered constrained control of State Dependent Parameter (SDP) models, a class of non–linear models with wide applicability to control problems. Research on NMSS–based control of SDP systems is still at an early stage and many issues are pending. The research in this thesis focused on constraint handling and stability.

The RG approach was initially considered for this class of non–linear systems, yielding similar conclusions as for the linear case in Chapter 3. Two different formulations of the MPC problem were subsequently considered, i.e. when the control action was directly utilised or parametrised using it's increments. For both problem formulations, a novel method to improve the predictions was proposed. In contrast to existing techniques in SDP-PIP control, the system was not considered *frozen* but rather the predicted future control action was used to predict the evolution of the system. The evolving system was used iteratively to obtain more accurate predictions of the system output and state. Based on these accurate predictions both control systems are shown to be asymptotically stable based on assumptions very similar to the case of linear systems.

The stability analysis (yet under the strong assumption that that the prediction of the system evolution has converged), represents the first stability result for control of SDP systems in this context. To the author's knowledge, it is the first

method that accounts for system constraints whilst defining a framework under which stable solutions can be achieved.

9.2 Directions for future research

Suggested directions for further research are summarised below.

9.2.1 Control structure

The importance of the control structure was highlighted at various places in the body of this thesis. However, in most cases only simulation results have been used to support the claims made. A theoretical justification has not yet been developed. More specifically, the Forward Path MPC controller resembles the Forward Path PIP controller and relative advantages and disadvantages are very similar in both cases. Clearly there must be an underlying theoretical reason for this, applicable to both MPC and PIP control.

The NMSS/MPC controller based on an integral–of–error state and the disturbance handling techniques developed in this thesis are more justified, in the sense that the difference lies in the system description or the parametrisation of the optimisation problem. However, consideration of further examples would be desirable and may lead to a deeper understanding of the different structures.

9.2.2 SDP/MPC control

As discussed above, research into SDP control systems is at an early stage. Therefore, there are various directions that could be followed outside the scope of the present thesis (e.g. relating to SDP/PIP control). However, some specific issues relating to the present research are listed below.

- The convergence of the algorithm that improves the predictions needs to be considered. There may be cases were a multiplicity in the solution may cause

the algorithm not to converge to a unique solution. It is therefore suggested that conditions under which the algorithm is covergent are sought.

- The time it takes for the improved solutions to be calculated has been assumed to be less than the sampling time. This may not always be the case and more detailed research on this issue should be carried out.

- Recent results on SDP/PIP control considered partial and exact linearisation by feedback of the SDP system. It would be interesting to consider this approach in the context of MPC systems and to combine it with the improvement in the predictions that has been presented here.

- There are various issues within the SDP/PIP research that should also be considered in the case of SDP/MPC control. For example, ill–conditioned cases (e.g. when the system becomes uncontrollable) have not been considered here. In addition, a duality in the solution can cause the algorithm that improves the prediction not to converge. Additional examples of badly behaved systems can be found in the references given in the main body of the thesis. Therefore it is suggested that more simulation examples could provide some insight towards this direction.

9.2.3 NMSS/MPC control applications

This thesis is based on new theoretical and methodological developments, supported by simulation rather than experimental results (with the exception of the modelling results in Chapter 7). It is therefore desirable in future work for the conclusions to be validated by application to real systems. A practical comparison of the different structures presented here can highlight the conclusions of this thesis or even expose issues that are not apparent from the simulations.

Finally, Chapter 5 considered tuning of MPC controllers. To the author's knowledge, this technique has not yet been applied in practical applications even

for the cases of PID and PIP controllers that are present in the literature. The results presented here should therefore be validated by practical examples to help gain a better understanding of the technique.

Appendices

Appendix A

Notation

A.1 Acronyms

CLP	Closed Loop Paradigm
dof.	Degrees of Freedom
EA	Evolutionary Algorithm
GA	Genetic Algorithm
GPC	Generalised Predictive Controller
KKT	Karush–Kuhn–Tucker
LQ	Linear Quadratic
MCS	Monte Carlo Simulation
MFD	Matrix Fraction Description
MIMO	Multi Input Multi Output
MPC	Model Predictive Control
NMSS	Non–Minimal State Space
PI	Proportional–Integral
PID	Proportional–Integral–Derivative
PIP	Proportional–Integral–Plus
QP	Quadratic Program

RG	Reference Governor
RHC	Receding Horizon Control
RIV	Refined Instrumental Variable
SDP	State Dependent Parameter
SISO	Single Input Single Output
SRIV	Simplified Refined Instrumental Variable
TVP	Time Variable Parameter
YIC	Young Identification Criterion

A.2 Nomenclature

q	Number of system inputs
p	Number of system outputs
z^{-1}	Backward shift operator
s	Laplace variable
Δ	Difference operator
$\mathbf{u}_k$	Input vector at kth sampling instant
$\mathbf{y}_k$	Output vector at kth sampling instant
$\mathbf{r}_k$	Reference vector at kth sampling instant
$\mathbf{w}_k$	Perturbed reference vector at kth sampling instant (RG scheme)
$\mathbf{v}_k$	Disturbance vector at kth sampling instant
$\mathbf{x}_k$	State vector at kth sampling instant
$\mathbf{z}_k$	Integral–of–error state at kth sampling instant
$\mathbf{A}$	System state matrix
$\mathbf{A}_k$	System state matrix at kth sampling instant
$\mathbf{B}$	System input matrix
$\mathbf{B}_k$	System input matrix at kth sampling instant

$\mathbf{D}$	System reference level matrix
$\mathbf{C}$	System output matrix
$a_i(\chi_{i,k})$	SDP output parameters at kth sampling instant
$b_i(\psi_{i,k})$	SDP input parameters at kth sampling instant
N_p	Prediction Horizon
N_c	Control Horizon
$\mathbf{Q}$, $\mathbf{R}$	Weighting matrices
$\mathbf{X}$	Vector of predicted states
$\mathbf{U}$	Vector of predicted control signals
$\underline{\mathbf{U}}$, $\overline{\mathbf{U}}$	Vectors of constraints on future control signals
$\mathbf{\Delta U}$	Vector of predicted control increments
$\underline{\mathbf{\Delta U}}$, $\overline{\mathbf{\Delta U}}$	Vectors of constraints on future control increments
$\mathbf{Y}$	Vector of predicted outputs
$\underline{\mathbf{Y}}$, $\overline{\mathbf{Y}}$	Vectors of constraints on future output signals
$\mathbf{S}$	Vector of future reference signals
$\underline{\mathbf{u}}$, $\overline{\mathbf{u}}$	Lower and upper bounds for the control vector
$\underline{\mathbf{\Delta u}}$, $\overline{\mathbf{\Delta u}}$	Lower and upper bounds for the control vector rate–of–change
$\underline{\mathbf{y}}$, $\overline{\mathbf{y}}$	Lower and upper bounds for the output vector
$\mathbf{p}$	Parameter vector
$\hat{\mathbf{p}}$	Estimated parameter vector
$\mathbf{P}^*$	Covariance matrix
R_T^2	Coefficient of determination

Appendix B

Convex sets and functions

In this appendix, some preliminaries and definitions are presented that are used in Section 2.2. A very useful property of sets and functions is convexity. Convex functions over convex sets lead to convex optimisation problems that are much easier to solve than general non–convex ones (Fletcher, 1981). Furthermore, although the results presented in the following hold for general optimisation problems, it is shown that the optimisation problem that will need to be solved in the rest of this thesis is in fact convex with some interesting properties. It has also been suggested (Maciejowski, 2002, and references therein) that by exploiting these properties more efficient algorithms can be used, a brief description of two of which is made in Sections 2.2.3 and 2.2.4.

B.1 Convex Sets

Definition B.1.1 (Convex set (Goodwin et al., 2005)). *A set $\mathcal{C} \subset \Re^n$ is convex if the line segment joining any two points of the set also belongs to the set, i.e. for every $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$ it follows that $\mathbf{x}_\theta \in \mathcal{C}$, where $\mathbf{x}_\theta = (1-\theta)\mathbf{x}_1 + \theta\mathbf{x}_2$ and $\theta \in [0,1]$.*

It is easy to show, by direct application of Definition B.1.1 that the following sets are convex.

- **Hyperplane.** $\mathcal{C} = \{\mathbf{x} : \mathbf{p}^T\mathbf{x} = \alpha\}$, where $\mathbf{p}$ is a nonzero vector in $\Re^n$ and α a scalar.

- **Polyhedral set.** $\mathcal{C} = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$, where $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{b}$ a vector of size $m \times 1$.

They are both very important in the MPC framework since they can represent linear constraints on system parameters and will be used extensively in the following.

The next two theorems provide convexity results on combinations of convex sets, while Lemma B.1.4 delivers a result that will later be used to prove convexity of the MPC optimisation problem.

Theorem B.1.2 (Convexity of linear combinations of sets (Luenberger, 1969)). *If $\mathcal{C}_1$ and $\mathcal{C}_2$ are convex sets then:*

- *The set $a\mathcal{C}_1 = \{\mathbf{x} : \mathbf{x} = a\mathbf{x}_1, \mathbf{x}_1 \in \mathcal{C}_1\}$ is convex.*

- *The set $\mathcal{C}_1 + \mathcal{C}_2 = \{\mathbf{x} : \mathbf{x} = \mathbf{x}_1 + \mathbf{x}_1, \mathbf{x}_1 \in \mathcal{C}_1$ and $\mathbf{x}_2 \in \mathcal{C}_2\}$ is convex.*

Theorem B.1.3 (Convexity of intersection of sets (Fletcher, 1981)). *If $\mathcal{C}_i$, $i = 1, 2, \ldots, m$ are convex sets then their intersection $\mathcal{C} = \bigcap_{i=1}^{m} \mathcal{C}_i$ is also a convex set.*

Lemma B.1.4. *The intersection of a number of hyperplanes and polyhedral sets is convex.*

Proof. The proof is a direct result of Theorem B.1.3 and the fact that both the hyperplane and the polyhedral sets are convex. □

B.2 Convex Functions

Next, various types of convexity of functions are defined.

Definition B.2.1 (Various types of Convexity of functions (Goodwin et al., 2005)). *Let $f : \mathcal{C} \rightarrow \Re$, where $\mathcal{C}$ is a nonempty set in $\Re^n$.*

- *f is **convex** on the convex set $\mathcal{C}$ if*

$$f(\theta \mathbf{x}_1 + (1-\theta)\mathbf{x}_2) \leq \theta f(\mathbf{x}_1) + (1-\theta) f(\mathbf{x}_2)$$

for each $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$ and for each $\theta \in (0,1)$.

The function f is strictly convex on $\mathcal{C}$ if the above inequality is true as a strict inequality for each distinct $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$ and for each $\theta \in (0,1)$.

- *f is **quasiconvex** on the convex set $\mathcal{C}$ if*

$$f(\theta \mathbf{x}_1 + (1-\theta)\mathbf{x}_2) \leq \max\{f(\mathbf{x}_1), f(\mathbf{x}_2)\}$$

for each $\mathbf{x}_1$, $\mathbf{x}_2 \in \mathcal{C}$ and $\theta \in (0,1)$.

The function f is strictly quasiconvex on $\mathcal{C}$ if the above inequality is true as a strict inequality for each $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$ with $f(\mathbf{x}_1) \neq f(\mathbf{x}_2)$.

- *a differentiable on $\mathcal{C}$ function f is **pseudoconvex** on $\mathcal{C}$ if $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$*

$$\nabla f(\mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1) \geq 0 \Rightarrow f(\mathbf{x}_2) \geq f(\mathbf{x}_1)$$

or alternatively

$$f(\mathbf{x}_2) < f(\mathbf{x}_1) \Rightarrow \nabla f(\mathbf{x}_1)(\mathbf{x}_2 - \mathbf{x}_1) < 0$$

The function f is strictly pseudoconvex on $\mathcal{C}$ if $f(\mathbf{x}_1) > f(\mathbf{x}_2)$ on the first condition or equivalently $f(\mathbf{x}_2) \leq f(\mathbf{x}_1)$ on the second condition.

Note that from definition a convex function is both quasiconvex and (under differentiability) pseudoconvex.

The following are two convex functions on $\Re^n$ that will be used later:

- **Linear function.** $f(\mathbf{x}) = \mathbf{p}^T\mathbf{x}$, where $\mathbf{p}$ is a vector in $\Re^n$.

- **Quadratic function.** $f(\mathbf{x}) = \mathbf{x}^T\mathbf{H}\mathbf{x}$, where $\mathbf{H}$ is a positive definite matrix, i.e. $\mathbf{H} \succ 0$ in $\Re^{n\times n}$.

They are both very important in the MPC framework since their combination constitutes the cost function. In the following, a theorem for convexity of sum of functions is given and Lemma B.2.3 delivers a result that will later be used to show that the MPC problem is convex.

Theorem B.2.2 (Convexity of sum of functions (Fletcher, 1981)). *If $f_i(\mathbf{x})$, $i = 1, 2, \ldots, m$ are convex functions on a convex set C, and $\alpha_i \geq 0$ then $\sum_{i=1}^{m} \alpha_i f_i(\mathbf{x})$ is also a convex function on C.*

Lemma B.2.3. *The sum of a linear and a quadratic function is convex on $\Re^n$.*

Proof. The proof is a direct result of Theorem B.2.2 and the fact that both the linear and quadratic functions are convex on $\Re^n$. □

Appendix C

Stability analysis

This appendix presents stability results for the general MPC control problem. There are various approaches to assess stability of MPC, a review of which is performed in Mayne et al. (2000). Most of them require a terminal cost or a terminal constraint, while Limon et al. (2006) have recently presented a suboptimal controller the makes the use of a terminal constraint obsolete (still using a terminal cost though). In the same direction, the dual mode MPC controller (Michalska and Mayne, 1993) requires the state to lie within a terminal set and not vanish to zero at the end of the prediction horizon.

C.1 Optimisation problem

Let the control action be the solution to the following optimisation problem.

$$\min_{\mathbf{u}_k,\ldots,\mathbf{u}_{k+N_c-1}} \quad \sum_{i=1}^{N_p} \mathbf{x}_{k+i|k}^T \mathbf{Q} \mathbf{x}_{k+i|k} + \sum_{i=0}^{N_c-1} \mathbf{u}_{k+i|k}^T \mathbf{R} \mathbf{u}_{k+i|k} \tag{C.1a}$$

$$\text{subject to:} \begin{cases} \underline{\mathbf{u}} \leq \mathbf{u}_{k+i} \leq \overline{\mathbf{u}} & , i = 0, \ldots, N_c - 1 \\ \underline{\boldsymbol{\Delta}\mathbf{u}} \leq \boldsymbol{\Delta}\mathbf{u}_{k+i} \leq \overline{\boldsymbol{\Delta}\mathbf{u}} & , i = 0, \ldots, N_c - 1 \\ \underline{\mathbf{y}} \leq \mathbf{y}_{k+i} \leq \overline{\mathbf{y}} & , i = 1, \ldots, N_p \end{cases} \tag{C.1b}$$

where $\mathbf{x}_{k+i|k}$ is the i-step ahead predicted value of a variable $\mathbf{x}$ conditional on sample k; $\boldsymbol{u}_{k+i|k}$ is the control vector at sample $k+i$ conditional on sample k; $\boldsymbol{\Delta}\mathbf{u}_{k+i|k}$, $i = 0, 1, \ldots, N_c - 1$ is the control increment vector at sample $k+i$ conditional on sample k; N_p and N_c are the *prediction* and *control* horizons respectively; the notation $\underline{\cdot}$ and $\bar{\cdot}$ represents vectors consisting of the minimum and maximum allowed values of a vector; and $\leq$ refers to element wise inequalities. Finally, $\mathbf{Q}$ and $\mathbf{R}$ are $p \times p$ and $q \times q$ weighting matrices chosen by the designer.

C.2 Lyapounov stability

In this analysis, a terminal constraint is used and it is clear that it does not differ from normal stability analysis for other MPC control schemes (e.g. Mayne et al., 2000), but is still presented here for completeness. In this regard Lyapounov's theorem (see Section 2.4) is evaluated to prove stability of the origin (it is evident that with the appropriate coordinate transformation, any equilibrium can be moved to the origin) under the following assumptions:

Assumption C.2.1. *An additional constraint is introduced to the constrained optimisation problem* (C.1). *Namely the predicted state after N_p samples is required to be to zero (i.e. $x_{k+N_p} = 0$) when the optimised control sequence has been applied to the system.*

Assumption C.2.2. *The constrained optimisation problem* (C.1) *is feasible at every sampling instant (with the additional terminal constraint introduced in Assumption C.2.1).*

The above assumptions are sufficient to prove stability for the described control scheme. This is summarised in the following Theorem.

Theorem C.2.1. *Under the Assumptions C.2.1 and C.2.2 the Model Predictive Control scheme that is derived by solving the constrained optimisation problem* (C.1) *is asymptotically stable.*

Proof. The minimum of the finite horizon cost function (C.1a) is chosen as a candidate Lyapounov function:

$$V(\mathbf{x}_k, \mathbf{u}^*_{|k\rightarrow}) = \sum_{i=1}^{N_p} \mathbf{x}^T_{k+i|k}\mathbf{Q}\mathbf{x}_{k+i|k} + \sum_{i=0}^{N_c-1} \mathbf{u}^{*T}_{k+i|k}\mathbf{R}\mathbf{u}^*_{k+i|k} \tag{C.2}$$

where $\mathbf{u}^*_{|k\rightarrow} = \left\{\mathbf{u}^*_{k+1|k}, \mathbf{u}^*_{k+2|k}, \ldots \mathbf{u}^*_{k+N_c-1|k}\right\}$ is the optimising sequence of the optimisation problem (C.1) at sampling instant k and the subscript $_{|k}$ is introduced to make explicit reference to the sampling instant that each prediction sequence refers to. It is easily seen that $V(\mathbf{x}, \mathbf{u})$ is positive definite and $V(\mathbf{x}, \mathbf{u}) = 0$ only if $(\mathbf{x}, \mathbf{u}) = (\mathbf{0}, \mathbf{0})$.

In the same manner, the value of the candidate Lyapounov function at the next sampling instant is:

$$V(\mathbf{x}_{k+1}, \mathbf{u}^*_{|k+1\rightarrow}) = \sum_{i=2}^{N_p+1} \mathbf{x}^T_{k+i|k+1}\mathbf{Q}\mathbf{x}_{k+i|k+1} + \sum_{i=1}^{N_c} \mathbf{u}^{*T}_{k+i|k+1}\mathbf{R}\mathbf{u}^*_{k+i|k+1}$$

A feasible solution (not necessarily optimal) to the optimisation problem (4.12) at sampling instant $k+1$ is the solution at the previous sampling instant (namely k) appended by a zero vector (i.e. $\mathbf{u}_{|k+1\rightarrow} = \left\{\mathbf{u}^*_{k+1|k}, \mathbf{u}^*_{k+2|k}, \ldots \mathbf{u}^*_{k+N_c-1|k}, \mathbf{0}\right\}$). Since this solution is not necessarily optimal, the following inequality holds:

$$V(\mathbf{x}_{k+1}, \mathbf{u}^*_{|k+1\rightarrow}) \leq \bar{V}(\mathbf{x}_{k+1}, \mathbf{u}_{|k+1\rightarrow})$$

where $\bar{V}$ is a suboptimal value of the cost:

$$\bar{V}(\mathbf{x}_{k+1}, \mathbf{u}_{|k+1\rightarrow}) = \sum_{i=2}^{N_p+1} \mathbf{x}^T_{k+i|k+1}\mathbf{Q}\mathbf{x}_{k+i|k+1} + \sum_{i=1}^{N_c} \mathbf{u}^T_{k+i|k+1}\mathbf{R}\mathbf{u}_{k+i|k+1}$$

It can now be shown that $V(\mathbf{x}, \mathbf{u})$ is decreasing by calculating the difference at

sampling instant k and $k+1$, i.e.

$$\begin{aligned}\Delta V(k+1) &= V(\mathbf{x}_{k+1}, \mathbf{u}^*_{|k+1\rightarrow}) - V(\mathbf{x}_k, \mathbf{u}^*_{|k\rightarrow}) \\ &\leq \bar{V}(\mathbf{x}_{k+1}, \mathbf{u}_{|k+1\rightarrow}) - V(\mathbf{x}_k, \mathbf{u}^*_{|k\rightarrow})\end{aligned}$$

Since the same control signal is applied to the model, the predictions at future sampling instants are identical (i.e. $\mathbf{x}_{k+i|k+1} = \mathbf{x}_{k+i|k}$, for $i = 2, \ldots, N_p$). Therefore, the difference can now be written as:

$$\Delta V(k+1) \leq \mathbf{x}^T_{k+N_p+1|k+1}\mathbf{Q}\mathbf{x}_{k+N_p+1|k+1} - \mathbf{x}^T_{k+1|k}\mathbf{Q}\mathbf{x}_{k+1|k} - \mathbf{u}^{*T}_{k|k}\mathbf{R}\mathbf{u}^*_{k|k} \qquad \text{(C.3)}$$

And from Assumption C.2.1 it follows

$$\begin{aligned}\Delta V(k+1) &\leq -\mathbf{x}^T_{k+1|k}\mathbf{Q}\mathbf{x}_{k+1|k} - \mathbf{u}^{*T}_{k|k}\mathbf{R}\mathbf{u}^*_{k|k} \\ &\leq 0\end{aligned}$$

That leads to the conclusion that $V(\mathbf{x}_{k+1}, \mathbf{u}^*_{|k+1\rightarrow}) \leq V(\mathbf{x}_k, \mathbf{u}^*_{|k\rightarrow})$, and according to Lyapounov's Theorem (Section 2.4) the system is asymptotically stable, which completes the proof. □

Remark C.2.1. *The introduction of Assumption C.2.1 forces the system to have settled at the end of the prediction horizon. Although this makes the proof of stability straightforward, it is possible to cause problems when solving the optimisation problem* (C.1)*. Especially in cases of large disturbances, the optimisation problem may become infeasible because of this additional equality constraint.*

References

B.M. Åkesson and H.T. Toivonen. State–dependent parameter modelling and identification of stochastic non–linear sampled–data systems. *Journal of Process Control*, 16(8):877–886, 2006.

R. K. Al Seyab and Y. Cao. Nonlinear model predictive control for the ALSTOM gasifier. *Journal of Process Control*, 16(8):795–808, 2006.

K.M. Anstreicher and R.M. Freund, editors. *Annals of Operation Research: Interior Point Methods in Mathematical Programming*, volume 62. Baltzer Science publishers, Amsterdam, 1996.

M. Bacic, M. Cannon, Y. I. Lee, and B. Kouvaritakis. General interpolation in MPC and its advantages. *IEEE Transactions on Automatic Control*, 48(6): 1092–1096, 2003.

T.A. Badgwell. Robust model predictive control of stable linear systems. *International Journal of Control*, 68(4):797–818, 1997.

R. A. Bartlett, L. T. Biegler, J. Beckstrom, and V. Gopal. Quadratic programming algorithms for large-scale model predictive control. *Journal of Process Control*, 12(7):775–795, 2002.

A. Bemporad. Model predictive control design: New trends and tools. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6678–6683, San Diego, California, USA, December 2006.

A. Bemporad. Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3):415–419, 1998a.

A. Bemporad. A predictive controller with artificial lyapunov function for linear systems with input/state constraints. *Automatica*, 34(10):1255–1260, 1998b.

A. Bemporad and M. Morari. Robust model predictive control: A survey. In A. Garulli, A. Tesi, and A. Vicino, editors, *Robustness in Identification and Control*. Springer Verlag, 1999.

A. Bemporad, A. Cassavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, 1997.

A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

L.T. Biegler. Advances in nonlinear programming concepts for process control. *Journal of Process Control*, 8(5):301–311, 1998.

H.H.J. Bloemen, T.J.J. van den Boom, and H.B. Verbruggen. Optimizing the end-point state-weighting matrix in model-based predictive control. *Automatica*, 38 (6):1061–1068, 2002.

F. Borrelli, M. Baotic, A. Bemporad, and M. Morari. Efficient on-line computation of constrained optimal control. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 1187–1192, Orlando, Florida, USA, December 2001.

X. Cheng and B.H. Krogh. Stability-constrained model predictive control. *IEEE Transactions on Automatic Control*, 46(11):1816–1820, 2001.

L. Chisci and E. Mosca. Stabilizing I–O receding horizon control of CARMA plants. *IEEE Transactions on Automatic Control*, 39(3):614–618, 1994.

L. Chisci and G. Zappa. Fast algorithm for a constrained infinite horizon LQ problem. *International Journal of Control*, 72(11):1020–1026, 1999.

A. Chotai, P. C. Young, P. McKenna, and W. Tych. PIP design for delta-operator systems: Part 2, MIMO systems. *International Journal of Control*, 70(1):149–168, 1998.

D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalised predictive control–Part I. The basic algorithm. *Automatica*, 23(2):137–148, 1987a.

D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalised predictive control–Part II. Extensions and interpretations. *Automatica*, 23(2):149–160, 1987b.

M. Diehl and J. Björnberg. Robust dynamic programming for min–max model predictive control of constrained uncertain systems. *IEEE Transactions on Automatic Control*, 49(12):2253–2257, 2004.

R. Dixon and A.W. Pike. Alstom benchmark challenge II on gasifier control. *IEE Proceedings: Control Theory and Applications*, 153(3):254–261, 2005.

A. Dunoyer, L. Balmer, K.J. Burnham, and D.J.G. James. On the discretization of single–input single–output bilinear systems. *International Journal of Control*, 68(2):361–372, 1997.

V. Exadaktylos, C.J. Taylor, and A. Chotai. Model predictive control using a non-minimal state space form with an integral-of-error state variable. In *UKACC International Conference on Control*, Glasgow, UK, August 2006.

P.J. Fleming and A.P. Pashkevich. Application of multi-objective optimisation to compensator design for SISO control systems. *Electronics Letters*, 22(5): 258–259, 1986.

P.J. Fleming and R.C. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10(11):1223–1241, 2002.

R. Fletcher. *Practical methods of optimization. Volume 2 Constrained optimisation.* Wiley, New York, 1981.

H. Fukushima and R.R. Bitmead. Robust constrained predictive control using comparison model. *Automatica*, 41(1):97–106, 2005.

A.F. Gilbert, A. Yousef, K. Natarajan, and S. Deighton. Tuning of PI controllers with one-way decoupling in 2×2 MIMO systems based on finite frequency response data. *Journal of Process Control*, 13(6):553–567, 2003.

P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, London, 1981.

G.C. Goodwin, M.M. Seron, and J.A. De Doná. *Constrained control and estimation: An optimisation approach.* Springer–Verlag, London, 2005.

P. Grieder, F. Borrelli, F. Torrisi, and M. Morari. Computation of the constrained infinite time linear quadratic regulator. *Automatica*, 40(4):701–708, 2004.

J. Gu, C.J. Taylor, and D. Seward. Proportional–integral–plus control of an intelligent excavator. *Computer-Aided Civil and Infrastructure Engineering*, 19(1): 16–27, 2004.

D. Guo and W.J. Rugh. A stability result for linear parameter-varying systems. *Systems & Control Letters*, 24(1):1–5, 1995.

K.J. Hunt and T.A. Johansen. Design and analysis of gain–scheduled control using local controller networks. *International Journal of Control*, 66(5):619–651, 1997.

L. Imsland, N. Bar, and B. A. Foss. More efficient predictive control. *Automatica*, 41(8):1395–1403, 2005.

T. Kailath. *Linear Systems.* Prentice-Hall, Englewood Cliffs, 1980.

R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the American Society of Mechanical Engineers–Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

E. C. Kerrigan and J. M. Maciejowski. Soft constraints and exact penalty functions in model predictive control. In *UKACC International Conference on Control*, Cambridge, UK, September 2000.

P. Kontoroupis, P.C. Young, A. Chotai, and C.J. Taylor. State Dependent Parameter-Proportional Intregral Plus (SDP-PIP) control of nonlinear systems. In *Proceedings 16th International Conference on Systems Engineering, ICSE 2003*, pages 373–378, Coventry University, 2003.

M.V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

B. Kouvaritakis, M. Cannon, and J.A. Rossiter. Non-linear model based predictive control. *International Journal of Control*, 72(10):919–928, 1999.

B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8):1545–1549, 2000.

B. Kouvaritakis, M. Cannon, and J.A. Rossiter. Who needs QP for linear MPC anyway? *Automatica*, 38(5):879–884, 2002.

W.H. Kwon and A.E. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Transactions on Automatic Control*, AC-22:838–842, 1977.

M.J. Lees. *Multivariable modelling and Proportional–Integral–Plus control of Greenhouse Climate.* PhD thesis, Lancaster University, 1996.

M.J. Lees, C.J. Taylor, A. Chotai, P.C. Young, and Z.S. Chalabi. Design and implementation of a proportional-integral-plus (PIP) control system for temperature, humidity and carbon dioxide in a glasshouse. *Acta Horticulturae (ISHS)*, 406: 115–123, 1996.

A.M. Letov. *Stability in nonlinear control systems.* Princeton University Press, 1961.

D. Limon, T. Alamo, F. Salas, and E.F. Camacho. On the stability of constraint MPC without terminal constraint. *IEEE Transactions on Automatic Control*, 51(5):832–836, 2006.

T. Liu, W. Zhang, and F. Gao. Analytical decoupling control strategy using a unity feedback control structure for MIMO processes with time delays. *Journal of Process Control*, 17(2):173–186, 2007.

D.G. Luenberger. *Optimisation by vector space methods.* Wiley, New York, 1969.

J.M. Maciejowski. *Predictive Control with Constraints.* Prentice–Hall, Hallow, England, 2002.

L. Magni. On robust tracking with non-linear model predictive control. *International Journal of Control*, 75(6):399–407, 2002.

T.E. Marlin. *Process Control: Designing Processes and Control systems for dynamic performance.* McGraw–Hill, New York, 1995.

D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2): 219–224, 2005.

D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: stability and optimality. *Automatica*, 36(6):789–814, 2000.

T.X. Mei and R.M. Goodall. LQG and GA solutions for active steering of railway vehicles. *IEE Proceedings: Control Theory and Applications*, 147(1):111–117, 2000.

A. Melman and R. Polyak. The newton modified barrier method for QP problems. In K.M. Anstreicher and R.M. Freund, editors, *Annals of Operation Research: Interior Point Methods in Mathematical Programming*, volume 62, pages 465–520. Baltzer Science publishers, Amsterdam, 1996.

H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

M. Morari. Model predictive control: Multivariable control of choice in the 1990? In D. Clarke, editor, *Advances in Model-Based Predictive Control*, pages 22–37. Oxford University Press, 1994.

E. Mosca. *Optimal predictive and adaptive control.* Prentice–Hall, Upper Saddle River, 1995.

K.R. Muske and T.A. Badgwell. Disturbance modelling for offset–free linear model predictive control. *Journal of Process Control*, 12(5):617–632, 2002.

S.-R. Oh and S.K. Agrawal. A reference governor based controller for a cable robot under input constraints. *IEEE Transactions on Control Systems Technology*, 13 (4):639–645, 2005.

C. Omnen, R. Babuška, U. Kaymak, J.M. Sousa, H.B. Verbruggen, and R. Isermann. Genetic algorithms for optimization in predictive control. *Control Engineering Practice*, 5(10):1363–1372, 1997.

G. Pannocchia and J.B. Rawlings. Disturbance models for offset–free model-predictive control. *American Institute of Chemical Engineers Journal*, 49(2): 426–437, 2003.

P.Y. Papalambros and D.J. Wilde. *Principles of Optimal Design: Modelling and Computation.* Cambridge University Press, 2000.

R.K. Pearson. Selecting nonlinear model structures for conputer control. *Journal of Process Control*, 13(1):1–26, 2003.

J.A. Primbs and V. Nevistić. Feasibility and stability of constrained finite receding horizon control. *Automatica*, 36(7):965–971, 2000.

S.J. Qin and T.A. Badgwell. An overview of industrial Model Predictive Control technology. In J.C. Kantor, C.E. García, and B. Carnahan, editors, *Proceedings of the 5th International Conference on Chemical Process Control*, volume 93 of *AIChE Symposium Series*, pages 232–256, Tahoe City, CA, 1996.

S. Quanten, P. McKenna, A. Van Brecht, A. Van Hirtum, P.C. Young, K. Janssens, and D. Berckmans. Model-based PIP control of the spatial temperature distribution in cars. *International Journal of Control*, 76(16):1628–1634, 2003.

C.V. Rao and J.B. Rawlings. Linear programming and model predictive control. *Journal of Process Control*, 10(2):283–289, 2000.

C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of interior point methods to model predictive control. *Journal of Optimization Theory and Applications*, 99(3):723–757, 1998.

J. B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, 1993.

M. A. Rodrigues and D. Odloak. Robust MPC for systems with output feedback and input saturation. *Journal of Process Control*, 15(7):837–846, 2005.

J.A. Rossiter. *Model-Based Predictive control: A practical approach.* CRC–Press, Boca Raton, 2003.

J.A. Rossiter and P. Grieder. Using interpolation to improve efficiency of multi-parametric predictive control. *Automatica*, 41(4):637–643, 2005.

A. Sanchez and M.R. Katebi. Predictive control of dissolved oxygen in an activated sludge wastewater treatment plant. In *Proceedings of the European Control Conference ECC'2003*, Cambridge, UK, 2003.

D.J. Sandoz, M.J. Desfoges, B. Lennox, and P.R. Goulding. Algorithms for inductrial model predictive control. *Computing & Control Engineering Journal*, 11 (3):125–134, 2000.

P. O. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *American Institute of Chemical Engineers Journal*, 45(8):1649–1659, 1999.

P.O.M. Scokaert, D.Q. Mayne, and J.B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654, 1999.

J.S. Shamma and D. Xiong. Set–valued methods for linear parameter varying systems. *Automatica*, 35(6):1081–1089, 1999.

M.A. Stables and C.J. Taylor. Non–linear control of ventilation rate using state–dependent parameter models. *Biosystems Engineering*, 95(1):7–18, 2006.

D.J. Stilwell and W.J. Rugh. Stability and $\mathcal{L}_2$ gain properties of LPV systems. *Automatica*, 38(10):1601–1606, 2002.

C. J. Taylor and E. M. Shaban. Multivariable proportional-integral-plus (PIP) control of the ALSTOM nonlinear gasifier simulation. *IEE Proceedings: Control Theory and Applications*, 153(3):277–285, 2006.

C. J. Taylor, A. Chotai, and P. C. Young. Proportional-integral-plus (PIP) control of time delay systems. *IMechE Proceedings: Journal of Systems and Control Engineering*, 212:37–48, 1998.

C. J. Taylor, P. C. Young, and A. Chotai. State space control system design based on non-minimal state-variable feedback : Further generalisation and unification results. *International Journal of Control*, 73:1329–1345, 2000a.

C. J. Taylor, A. Chotai, and P. C. Young. Design and application of PIP controllers: robust control of the IFAC'93 benchmark. *Transactions of the Institute of Measurement and Control*, 23:183–200, 2001.

C.J. Taylor. Pole assignment control of state dependent parameter models: further examples and generalisations. Technical report, Lancaster University, 2005.

C.J. Taylor, P.C. Young, A. Chotai, W. Tych, and M.J. Lees. The importance of structure in PIP control design. In *UKACC International Conference on Control, Institute of Electrical Engineering, Conference Publication no. 427*, volume 2, pages 1196–1201, Exeter, UK, 1996.

C.J. Taylor, P.C. Young, A. Chotai, A.R. McLeod, and A.R. Glasock. Modelling and proportional-integral-plus control design for air-free carbon dioxide enrichment systems. *Journal of Agricultural Engineering Research*, 75(4):365–374, 2000b.

C.J. Taylor, P. Leigh, L. Price, P.C. Young, E. Vranken, and D. Berckmans. Proportional-integral-plus (PIP) control of ventilation rate in agricultural buildings. *Control Engineering Practice*, 12(2):225–233, 2004a.

C.J. Taylor, P.A. Leigh, A. Chotai, P.C. Young, E. Vranken, and D. Berckmans. Cost effective combined axial fan and throttling valve control of ventilation rate. *IEE Proceedings: Control Theory and Applications*, 151(5):577–584, 2004b.

C.J. Taylor, E.M. Shaban, A. Chotai, and S. Ako. Nonlinear control system design for construction robots using state dependent parameter models. In *UKACC International Conference on Control*, Glasgow, UK, August 2006a.

C.J. Taylor, E.M. Shaban, A. Chotai, and S. Ako. Development of an automated verticality alignment system for a vibro–lance. In *7th Portuguese Conference on Automatic Control*, Lisbon, Portugal, September 2006b.

H.T. Toivonen. State–dependent parameter models of non–linear sampled–data systems: a velocity–based linearization approach. *International Journal of Control*, 76(18):1823–1832, 2003.

H.T. Toivonen, S. Tötterman, and B. Åkesson. Identification of state–dependent parameter models with support vector regression. *International Journal of Control*, 80(1):1–17, 2007.

W. Tych. Multiobjective optimisation technique for mapping technical design objectives into the values of weighting matrices in the linear quadratic regulator design problem. Technical Report TR–117, CRES, Lancaster University, 1994.

W. Tych, P.C. Young, A. Chotai, and L.C. Jones. TDC: Computer aided true digital control of multivariable delta operator systems. In *Proceedings of the 13th IFAC World Congress*, paper No. 5c-01 5, San Francisco, July 1996.

A. Van Brecht, S. Quanten, T. Zerihundesta, S. Van Buggenhout, and D. Berckmans. Control of the 3–D spatio–temporal distribution of air temperature. *International Journal of Control*, 78(2):88–99, 2005.

C. Vlachos, D. Williams, and J.B. Gomm. Genetic approach to decentralised PI controller tuning for multivariable processes. *IEE Proceedings: Control Theory and Applications*, 146(1):58–64, 1999.

C. Vlachos, D. Williams, and J.B. Gomm. Solution to the Shell standard control

problem using genetically tuned PID controllers. *Control Engineering Practice*, 10(2):151–163, 2002.

C.-L. Wang and P.C. Young. Direct digital control by input–output, state variable feedback: theoretical background. *International Journal of Control*, 47(1):97–109, 1988.

L. Wang and P.C. Young. An improved structure for model predictive control using non-minimal state space realisation. *Journal of Process Control*, 16(4): 355–371, 2006.

Q.-G. Wang, B. Huang, and X. Guo. Auto-tuning of TITO decoupling controllers from step tests. *ISA Transactions*, 39(4):407–418, 2000.

Q.-G. Wang, Y. Zhang, and M.-S. Chiu. Decoupling internal model control for multivariable systems with multiple time delays. *Chemical Engineering Science*, 57(1):115–124, 2002.

Y. J. Wang and J. B. Rawlings. A new robust model predictive control method I: theory and computation. *Journal of Process Control*, 14(3):231–247, 2004.

J. F. Whidborne, G. Murad, D. W. Gu, and I. Postlethwaite. Robust control of an unknown plant - the IFAC 93 benchmark. *International Journal of Control*, 61(3):589–640, 1995.

P. C. Young. An instrumental variable approach to ARMA model identification and estimation. In *Proceedings 14th IFAC Symposium on System Identification SYSID06*, pages 410–415, Newcastle, NSW, Australia, 2006.

P. C. Young. Some observations on instrumental variable methods of time-series analysis. *International Journal of Control*, 23:593–612, 1976.

P. C. Young. *Recursive estimation and time-series analysis.* Springer–Verlag, 1984.

P. C. Young. Simplified refined instrumental variable (SRIV) estimation and true digital control (TDC): a tutorial introduction. In *1st European Control Conference*, pages 1295–1306, Grenoble, 1991.

P. C. Young. Data-based mechanistic modelling, generalised sensitivity and dominant mode analysis. *Computer Physics Communications*, 117:113–129, 1999.

P.C. Young. Applying parameter estimation to dynamic systems: Part I, theory. *Control Engineering*, 16:119–125, 1969.

P.C. Young, M.A. Behzadi, C.L. Wang, and A. Chotai. Direct digital and adaptive control by input–output state variable feedback pole assignment. *International Journal of Control*, 46(6):1867–1881, 1987.

P.C. Young, A. Chotai, P. McKenna, and W. Tych. Pip design for delta-operator systems: Part 1, SISO systems. *International Journal of Control*, 70(1):123–147, 1998.

P.C. Young, P. McKenna, and J. Bruun. Identification of non-linear stochastic systems by state dependent parameter estimation. *International Journal of Control*, 74(18):1837–1857, 2001.

S.H. Żak. *Systems and Control.* Oxford University Press, 2003.