Universitat Autònoma de Barcelona

Robustness aspects of Model Predictive Control

A DISSERTATION SUBMITTED IN
PARTIAL FULFILMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY AT THE
UNIVERSITAT AUTÒNOMA DE BARCELONA

David Megías Jiménez

March 2000

Dr. Javier Serrano García, lecturer at the Universitat Autònoma de Barcelona,

CERTIFIES:

that this dissertation, submitted by David Megías Jiménez in partial fulfilment for the degree of Doctor of Philosophy, has been written under his supervision.

Dr. Javier Serrano García

Bellaterra, March 2000

"Cuius rei demonstrationem mirabilem sane detex hanc marginis exiguitas non caparet"

Pierre de Fermat

"The computer is being used in essentially the same way that the experimental physicist uses a piece of experimental apparatus to explore the structure of the physical world"

Roger Penrose

A la Sandra i

als meus pares

Robustness aspects of Model Predictive Control

David Megías Jiménez

A dissertation submitted in partial fulfilment for the degree of
Doctor of Philosophy at the Universitat Autònoma de Barcelona
March 2000

Abstract

Model, *Model-based* or *Receding-horizon Predictive Control* (MPC or RHPC) is a successful and mature control strategy which has gained the widespread acceptance of both academia and industry. The basis of these control laws, which have been reported to handle quite complex dynamics, is to perform predictions of the system to be controlled by means of a model. A control profile is then computed to minimise some cost function defined in terms the predictions and the hypothesised controls.

It was soon realised that the first few predictive controllers failed to fulfil essential properties, such as the *stability* of the nominal closed-loop system. In addition, it was noticed that the discrepancies between the model and the true process, referred to as system uncertainty, can seriously affect the achieved performance. The *robustness* problem should, thus, be addressed. In this thesis, the problems of nominal stability and robustness are reviewed and investigated. In particular, the *accomplishment of constraint specifications* in the presence of various sources of uncertainty is a major objective of the methods developed throughout this PhD research.

First of all, controllers which guarantee nominal stability, such as the CRHPC and the GPC^{∞}, are highlighted and formulated, and 1-norm counterparts are obtained. The robustness of these strategies in the unconstrained case has been analysed, and it has been concluded that the infinite horizon approach often leads to more convenient performance and robustness results for typical choices of the tuning knobs. Then the constrained case has been undertaken, and *min-max controllers* based on the *global uncertainty* approach have been formulated for both 1-norm and 2-norm formulations. For these methods, a *band updating algorithm* has been suggested to modify the assumed uncertainty bounds on-line. Although both formulations provide similar results, which overcome the classical approach to robustness when constraints are specified, the 1-norm controllers are computationally more efficient, since the optimal control move sequence can be computed with a standard LP problem.

Finally, a refinement of the min-max approach which includes the notion that feedback is present in the receding-horizon implementation of predictive controllers, termed as *feedback min-max* MPC, is shown to overcome some of the drawbacks of the standard min-max approach.

Acknowledgements

En clara referència a les paraules de Pierre de Fermat que s'han reproduït en les pàgines prèvies a aquests agraïments, la llista de les persones que, d'una manera o d'una altra, han contribuït o m'han ajudat en la realització d'aquesta tesi doctoral és massa extensa i no cap en aquestes línies. Per aquest motiu, vull demanar sinceres excuses a tots aquells qui no s'esmenten explícitament, i els faig arribar, també, el meu agraïment.

Vull agrair al Dr. Javier Serrano la direcció d'aquest treball. L'orientació, l'ajut i els ànims que m'ha proporcionat han estat decisius per a la finalització d'aquesta recerca.

També vull fer arribar el meu agraïment al Dr. Ignasi Serra qui, com a cap del grup de recerca, m'ha fet adquirir la confiança necessària per tal de tirar endavant aquesta feina. Aquesta gratitud la faig extensiva a la resta de membres de la Unitat d'Enginyeria de Sistemes i d'Automàtica, especialment als meus companys de despatx per la seva comprensió, sobre tot en els moments més foscos.

I must not forget to mention the cooperation of the people of the STC group at the University of Oxford. I'm very grateful with Dr. David W. Clarke and with Dr. Alexander G. Kuznetsov for their help and supervision during my visit to the Department of Engineering Science. I also acknowledge the support provided by the "chaps" who shared the lab time with me, since they contributed to make my visit more pleasant. Besides, I'd like to thank all the people of NOOC, the walks through the park by the new building, and the beers in the pubs which filled the gaps between the "lab hours".

Debo hacer también mención especial del Dr. César de Prada y de todo su grupo de investigación de la Universidad de Valladolid. Las aportaciones y los consejos allí recibidos han permitido llevar a cabo gran parte de esta investigación. Aquí quiero también destacar a Paco del Valle, por las interesantes charlas a la hora del café, y a Smaranda y a Teresa Álvarez por las molestias que les haya podido causar mi presencia en sus respectivos despachos.

A la Sandra vull agrair-li l'afecte, la comprensió i el suport incondicional, coses sense les quals aquest vaixell hauria naufragat molt abans d'arribar a bon port. També li demano perdó explícitament per haver-me sentit renegar i queixar-me en els moments difícils, i per haver compartit moltes de les meves frustracions. Moltes gràcies, Sandra, pels millors moments de tot aquest temps.

No menys importants que totes les persones ja esmentades són el conjunt que formen els meus amics i la seva família. Per tòpic que pugui semblar, sense ells no hauria estat capaç ni de començar aquesta aventura.

Vull agrair el suport econòmic del Comissionat per a Universitats i Recerca (CIRIT) de la Generalitat de Catalunya que, mitjançant la beca 1996BEAI200179, em va permetre realitzar una estada de sis mesos a Oxford.

También quiero agradecer la financiación del proyecto TAP97-1144 por parte de la Comisión Interministerial de Ciencia y Teconología (CICYT), que también ha contribuido al desarrollo de esta tesis.

Contents

List of Principal Symbols

A Abbreviations and acronyms

2-DOF	Two-Degrees-Of-Freedom
APCS	Adaptive Predictive Control Systems
BIBO	Bounded-Input/Bounded-Output
CARIMA	Controlled Auto-Regressive Integrated Moving Average
CARMA	Controlled Auto-Regressive Moving Average
CPU	Central Processing Unit
CRHPC	Constrained Receding-Horizon Predictive Control (2-norm)
$CRHPC_1$	1-norm CRHPC
DMC	Dynamic Matrix Control
EHAC	Extended Horizon Adaptive Control
EPSAC	Extended Prediction Self-Adaptive Control
FIR	Finite Impulse Response
GPC	Generalised Minimum Variance
GPC	Generalised Predictive Control (2-norm)
GPC_1	1-norm GPC

GPC^{∞}	Infinite horizon GPC (2-norm)
GPC_1^{∞}	1-norm GPC^{∞}
IMC	Internal Model Control
IDCOM	Identification-Command
IHSPC	Infinite Horizon Stable Predictive Control
LP	Linear Programming
LQ, LQG	Linear Quadratic, Linear Quadratic Gaussian
LTI	Linear and Time Invariant
MAC	Model Algorithmic Control
MBPC	Model-Based Predictive Control
MHz	Mega Hertz (a million operations per second)
MIMO	Multi-Input/Multi-Output
MPC	Model-based Predictive Control/Model Predictive Control
MUSMAR	Multi-Step Multi-variable Adaptive Regulator
NLPC	Non-Linear Predictive Control
NP	Nominal Performance
NS	Nominal Stability
PFC	Predictive Functional Control
PID	Proportional plus Integral plus Derivative
QP	Quadratic Programming
$QGPC_1^{\infty}$	Quasi-infinite horizon 1-norm GPC
RP	Robust Performance

RS	Robust Stability
RHPC	Receding-Horizon Predictive Control
SGPC	Stable Generalised Predictive Control
SISO	Single-Input/Single Output
SIORHC	Stabilising Input/Output Receding-Horizon Control
SQP	Sequential Quadratic Programming
YKPC	Youla-Kučera Predictive Control
ZOH	Zero-Order Hold

B Predictive control terminology

i, j, k	integer indices
u, $u(t)$	input signal
y, $y(t)$	process output
$\tilde{y}$, $\tilde{y}(t)$	output of the unstable part of the process
w, $w(t)$	setpoint (or reference) signal
$\tilde{w}$, $\tilde{w}(t)$	setpoint of the unstable part of the process
f, $f(t)$	free response signal
$\tilde{f}$, $\tilde{f}(t)$	free response of the unstable part of the process
ξ, $\xi(t)$	zero-mean stochastic disturbance signal
e, $e(t)$	error signal ($e = w - y$)
J_n, $J_n(t)$	n-norm cost function
J_n^{opt}, $J_n^{\text{opt}}(t)$	optimal cost value (unconstrained case)

$\overline{J_n^{\text{opt}}}$, $\overline{J_n^{\text{opt}}}(t)$	optimal cost value (constrained case)
$\Delta\boldsymbol{u}$, $\Delta\boldsymbol{u}(t)$	hypothesised control move vector
$\Delta\boldsymbol{U}$, $\Delta\boldsymbol{U}(t)$	extended hypothesised control move vector
$\Delta\boldsymbol{u}^{\text{opt}}$, $\Delta\boldsymbol{u}^{\text{opt}}(t)$	optimal postulated control move vector (unconstrained case)
$\overline{\Delta\boldsymbol{u}^{\text{opt}}}$, $\overline{\Delta\boldsymbol{u}^{\text{opt}}}(t)$	optimal postulated control move vector (constrained case)
$\overline{\Delta\boldsymbol{U}^{\text{opt}}}$, $\overline{\Delta\boldsymbol{U}^{\text{opt}}}(t)$	optimal extended postulated control move vector (constrained case)
N_1	lower costing or prediction horizon
N_y	lower output constraint horizon
N_2	upper costing or prediction horizon
N_u	control horizon
N	prediction horizon (specific to CRHPC and GPC^{∞})
m	number of end-point equality constraints (specific to CRHPC)
μ, $\mu(j)$	tracking error weighting in cost function
ρ, $\rho(j)$	control effort weighting in cost function
A, $A(q^{-1})$	internal denominator polynomial
$\bar{A}$, $\bar{A}(q^{-1})$	stable poles of the internal denominator polynomial
$\widetilde{A}$, $\widetilde{A}(q^{-1})$	unstable poles of the internal denominator polynomial
$q^{-1}B$, $q^{-1}B(q^{-1})$	internal numerator polynomial
T, $T(q^{-1})$	observer or noise polynomial
P_c, $P_c(q^{-1})$	nominal closed-loop characteristic polynomial
P_{c0}, $P_{c0}(q^{-1})$	true closed-loop characteristic polynomial
g_j	j^{th} process step response coefficient

$\tilde{g}_j$	j^{th} step response coefficient of the unstable part of the process
x, $x(t)$	additive disturbance
θ, $\theta(t)$	global uncertainty

C Useful operators and standard symbols

$(t+j\|t)$	prediction for time $t+j$ subject to information available at time t. In the case of control moves, postulated control move for time $t+j$ to be computed at time t
q^{-1}	backward shift operator
Δ	differencing operator ($\Delta = 1 - q^{-1}$)
s	Laplace operator
A, $A(q^{-1})$	polynomial in terms of q^{-1}
a_j	q^{-j} coefficient of the polynomial A
$\deg(A)$	order of the polynomial A
$\boldsymbol{a}$	general vector
$\dim(\boldsymbol{a})$	dimension (or size) of the vector $\boldsymbol{a}$
$\\|\boldsymbol{a}\\|_n$	n-norm of $\boldsymbol{a}$ (Euclidean norm of $\boldsymbol{a}$ if n is omitted)
$\mathbf{1}$	ones vector of appropriate dimension
$\boldsymbol{A}$	general matrix
$\boldsymbol{I}$	identity matrix of appropriate dimension
$\boldsymbol{I}_n$	$n \times n$ identity matrix
$\mathbf{0}$	zero matrix or vector of appropriate dimension
$\mathbf{0}_{n,m}$	zero matrix of dimension $n \times m$

$\|\boldsymbol{A}\|_n$	induced n-norm of $\boldsymbol{A}$ (induced Euclidean norm of $\boldsymbol{A}$ if n is omitted)
$\jmath$	$\sqrt{-1}$
Δ_a	additive uncertainty
Δ_{ia}	inverse additive uncertainty
Δ_m	multiplicative uncertainty
Δ_{im}	inverse multiplicative uncertainty

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to Model Predictive Control

Model, *Model-based* or *Receding-horizon Predictive Control* (MPC or RHPC) is a successful control strategy which emerged in the late 1970's to face some industrial control problems. The basis of these methods is to make use of an explicit *model* of the system to be controlled in order to perform output and/or state *predictions*. An optimal control profile is computed to minimise a *cost or objective function* defined in terms of the predicted outputs (or states) and the control moves (or signals) over a given *prediction or coincidence horizon or window*.

In the last two decades, a large variety of methods have been developed within the MPC family and countless successful industrial applications have been reported, some of which can be found in (Qin and Badgwell, 1996). But the success of MPC is by no means confined to the industrial domain. Innumerable scientific publications, some of which are referenced hereafter, evidence the interest of academia in this control strategy. Some reasons to justify the attention paid to these methods from both the academical and the industrials worlds are given below:

1. MPC controllers have been developed either for linear or non-linear models.

2. There are no conceptual differences between *Single-Input/Single-Output* (SISO) and *Multiple-Input/Multiple-Output* (MIMO) formulations. In the latter case, the interaction between variables is compensated.

3. Difficult dynamics such as dead-times, unstable or non-minimum phase systems can be easily handled.

4. Feedforward compensation of measurable disturbances can be introduced in a natural way exploiting the model-based and predictive features of the MPC methodology by using disturbance models.

5. The incorporation of constraints in the manipulated and controlled variables and/or states is a simple task. Constraints can be considered at the controller design stage and the resulting optimisation problem can often be solved using standard *Linear Programming* (LP) or *Quadratic Programming* (QP) tools.

6. Preprogrammed setpoints, typical in robotics or batch processes, can be introduced.

7. Methods which guarantee the stability of the closed-loop system are available.

8. Robustness features can be enhanced through tuning parameters or optimisation methods.

Constraint handling is, indeed, one of the most appealing properties of MPC, since limits of several kinds always occur in practice. Constraints can be used to describe security limits (the pressure within a chemical reactor must be below some critical value), physical restrictions (a valve cannot be opened beyond a 100%), technological requirements (the temperature of a given process must be kept between some bounds), product quality specifications (maximum impurity allowed), and so on. These requirements must be handled at the controller design stage to avoid undesirable performance.

The objective of constraints is twofold. On the one hand, they can be used to increase the accuracy of the model, since actuator and plant limits can be incorporated into the model. On the other hand, constraints can be used as tuning knobs to describe control requirements or specifications. Usually, the optimal operating point lies close to (or on) one or several limits and therefore, from an economical point of view, it is advisable to operate as close to the constraint boundary as possible. If constraints are incorporated at the controller design stage, the operating point can often be specified much closer to the optimal location.

Constraints can be classified according to different criteria. The following classification of constraints, according to practical considerations, is due to Álvarez and de Prada (1997):

1. *Physical constraints.* These limits, which must never be surpassed, are determined by the physical limitations of the system.

2. *Operating constraints.* These bounds are fixed by the plant operators to specify the optimal operating region. The operation constraints are more restrictive than the physical limits.

3. *Optimisation or setpoint conditioning constraints.* These limits, more restrictive than the operating constraints, are used only if the setpoint conditioning technique is applied.

4. *Working constraints.* These are the actual constraints considered by the controller to determine the feasible region. The working constraints are obtained by choosing the most restrictive among the physical, the operating and the optimisation limits.

Apart from constraint handling, the relevant issues of *stability* and *robustness* have been successfully tackled in the last decade. The first few controllers in the MPC family did not guarantee the stability of the closed-loop system even when the true process and the assumed model were identical. This flaw was promptly overcome and several stabilising controllers have furnished the MPC class. At the same time, the robustness issue has been undertaken from several points of view. The process model is a key parameter of predictive controllers, and thus it must be analysed how the modelling errors and disturbances (system uncertainty) affect the closed-loop system. It would be hazardous to apply the controller directly to the true system if the sources of uncertainty are not carefully examined and quantified. Uncertainty might not only spoil performance and lead to constraint violations, but even instability could arise.

1.2 Historical overview of MPC

Some of the ideas which originate the first few predictive controllers were adopted from optimal control methodologies, such as the Linear Quadratic (LQ) or the Linear Quadratic Gaussian (LQG) controllers (Anderson and Moore, 1971; Kwakernnak and Sivan, 1972), but it was not until the late 1970's that the first two purely MPC control laws emerged. The *Identification-Command* (IDCOM) and the *Dynamic Matrix Control* (DMC), detailed in (Richalet *et al.*, 1978) and (Cutler and Ramaker, 1980) respectively, are acknowledged as the roots of MPC. These two methods share some common features which established the basis of MPC. To begin with, a dynamical model (the impulse response in the former and the step response in the latter) is explicitly used to assess the effect of the hypothesised future control actions. A control profile is thus computed to minimise a cost function which includes the setpoint tracking error besides the control effort, subject to operating constraints. The basis of MPC can be summarised in the following four points (Scokaert, 1994; Serrano, 1994; Cristea, 1998):

1. A process model is used to predict the future behaviour of the system over a coincidence or costing horizon or window using past input/output data and a

hypothesised sequence of future controls.

2. An objective or cost function based on some performance criterion is minimised over the coincidence horizon. The cost function is usually defined as a combination of some norm of the tracking errors and the control effort.

3. The optimisation problem yields an "open-loop optimal" control move sequence.

4. Only the first component of this sequence is implemented and the loop is closed by repeating this procedure at each sampling instant updating the past data with system's measurements. This is the usual *receding-horizon* strategy common to most MPC laws.

Different choices or alternatives can be given for the steps 1 through 4, and thus a great deal of degrees of freedom are available to design a MPC control strategy. Each possible choice results on a different MPC controller. This multiplicity of possibilities has given rise to a plethora of MPC controllers, some of which are surveyed below.

Apart from the relation to the LQ and LQG controllers, some of the ideas behind MPC are reported in several other approaches. Zadeh and Whalen (1962) related the optimal control problem with LP methods. Propoi (1963) suggested the receding-horizon strategy and Chang and Seborg (1983) highlighted the link between the methods of Propoi and MPC. Other approaches such as the Smith predictor, feedforward control or the *Internal Model Control* (IMC), surveyed in (Garcia and Morari, 1982), establish a relation between signal-based control such as *Proportional plus Integral plus Derivative* (PID) with the strategies which use an explicit model on-line, *e.g.* MPC.

Since the early results of MPC in the late 1970's, the acceptance of these methodologies in the process industry has grown unceasingly. As an example, several applications were reported in the 1980's (Mehra *et al.*, 1982; Garcia and Morshedi, 1984; Matsko, 1985; Martin *et al.*, 1986; Cutler and Hawkins, 1987), and some predictive control packages were launched in that decade. In parallel to the developments in industry, various academical groups across Europe focused on MPC and the first few results were soon available. The "academical" algorithms were developed for SISO systems and not much attention was paid to the constraint issue, but the reformulation of these methods to handle MIMO systems and to incorporate constraints is often straightforward. Among these controllers are the *Model Algorithmic Control* (MAC) of Rouhani and Mehra (1982), the Predictor-based Self-Tuning Control of Peterka (1984), the *Adaptive Predictive Control Systems* (APCS) of Martín Sánchez and Rodellar (1996) the *Extended Horizon Adaptive Control* (EHAC) of Ydstie (1984), the *Multi-Step Multivariable Adaptive Regulator* (MUSMAR) of Mosca *et al.* (1984) and the *Extended Prediction Self-Adaptive Control* (EPSAC) of De Keyser and Cauwenberghe (1985).

In the late 1980's, the widely known *Generalised Predictive Control* (GPC) was suggested by Clarke *et al.* (1987). This method gained the early recognition of academia and several research groups focused on the GPC law. One of the key points to justify the successful irruption of this control scheme is the inclusion of some previous MPC strategies (*e.g.* the DMC) as particular cases of the GPC. The precursors of the GPC controller are the minimum variance controller described by Åström (1970) and the self-tuning regulator of Åström and Wittenmark (1973). Given a linear model, the minimum variance controller is obtained to minimise the output variance criterion:

$$J(t) = E\left\{[y(t+1) - w(t+1)]^2\right\},$$

where $E\{\cdot\}$ denotes the expected value, $y(t)$ stands for the system output and $w(t)$ is the setpoint (or reference) signal. Hence, $u(t)$ is computed as the value which provides the minimum of $J(t)$, and the same problem is solved at time $t+1$ to compute $u(t+1)$. For delays d greater than 1, $t+d$ replaces $t+1$ is used in the definition of $J(t)$.

It is widely known that this kind of strategy is only possible for minimum phase systems, since the minimum variance law cancels out the zeroes of the plant and, obviously, unstable zeroes cannot be directly cancelled preserving the internal stability of the feedback system. If a weighting factor is introduced in the control signal, this technique can be applied to non-minimum phase systems, leading to the criterion:

$$J(t) = E\left\{[y(t+d) - w(t+d)]^2 + \mu u^2(t)\right\},$$

referred to as *Generalised Minimum Variance* (GMV) controller. A usual modification to the GMV law is to consider the control move $\Delta u(t)$ instead of the control signal, *i.e.*

$$J(t) = E\left\{[y(t+d) - w(t+d)]^2 + \mu \Delta u^2(t)\right\}.$$

This approach yields offset-free setpoint tracking for constant setpoints even when the plant does not include an integrator. On the other hand, in the standard GMV definition, a non-zero control signal is penalised always, even when the setpoint is different from zero.

The GPC can be viewed as an extension of the GMV law conceived to overcome the stability problems of the latter. The GMV fails to stabilise some unstable or non-minimum phase systems, especially if the delay is an uncertain (or time varying) parameter. The GPC extends the cost function further in the future, using the the prediction or coincidence horizon concept likewise the DMC. The most celebrated feature of GPC is the ability of producing a convenient closed-loop behaviour for a wide class of typical systems with not too many tuning knobs. This property favoured the application of GPC in industrial control problems, and other MPC controllers followed the GPC's journey from academia to industry. A few examples of successful applications of "academical" MPC controllers are reported in (Richalet, 1993*a*; Camacho and Berenguel, 1994).

However, a deep analysis of the GPC law carried out by Bitmead *et al.* (1990) revealed a major drawback: the stability of the closed-loop system could not be guaranteed. This difficulty was joined by the open question of the influence of system uncertainty in the closed-loop behaviour. Undoubtedly, stability and robustness became the preferred fields of research within the MPC framework in the 1990's. The historical revision about these two issues is suspended here. This task is resumed in the introductory sections of Chapters 2, 3, 4, and 5, where the state of the art of the stability and robustness problems is reviewed.

Along with the results of stability and robustness, the use of MPC for non-linear systems captured the attention of academia. Non-linear systems can often be controlled with linear MPC, as shown for example in (Megías, 1994; Serrano *et al.*, 1994), but some non-linearities are too difficult to be handled with these controllers. In the 1990's the use of non-linear models to develop MPC laws was investigated. The results of Mayne and Michalska (1990), Michalska and Mayne (1993) and Chen and Allgöwer (1998*b*) contributed to this new area of MPC, and provide with methods which guarantee nominal stability. An exhaustive survey of non-linear MPC algorithms with stability guarantees is given in (Chen and Allgöwer, 1998*a*). The development of non-linear MPC is, however, quite different from the linear case. Whereas the first few linear predictive controllers were conceived in the industrial environment and then attracted the attention of academia, non-linear MPC is almost confined to the academical community and the application field is still unexplored. The main difficulty to use the non-linear MPC approach in practical control problems is the enormous computational burden it requires. This drawback limits the application domain of these new methods to very slow processes. In addition, some issues require further research, such as the robustness features of non-linear MPC schemes.

With the current technology, the application of non-linear MPC seems too ambitious. To overcome this difficulty, some methods have been suggested halfway from non-linear and linear MPC. These controllers make use of a non-linear process model to make predictions, but the model is linearised on-line at each sampling instant to cut down the computations. For example, the solutions suggested by Oliveira *et al.* (1995), El Ghoumari (1998), Oliveira and Morari (1998) and Megías *et al.* (1999*b*) have proved successful as the computational burden is concerned. Nevertheless, purely linear MPC is still dominant in the application area.

1.3 Objectives and structure of the thesis

The main aim of this thesis is to research the combined problem of constraint handling, stability, and robustness. That is, starting from methods which guarantee closed-loop stability when the model and the true process coincide, endow these MPC controllers with a "robustness layer" to allow for some degree of uncertainty. The robustness requirement is specified as follows:

1. Stability and constraint accomplishment must be preserved in spite of uncertainty.
2. All types of uncertainty (linear, non-linear, time invariant, time varying, stable, unstable, parametric, non-parametric, etc.) must be considered.

In addition, the methods presented below were requested to be computationally efficient so that they could be applied even when fast dynamics occur. Finally, the newly developed controllers must be contrasted with the existing solutions provided in the literature to evaluate the quality of the results.

The methods developed here are based on the GPC or in further evolutions of this predictive controller which guarantee stability. There are several reasons for such a choice. To begin with, many applications of the GPC have been reported, and it is thus convenient to undertake the stability and robustness problems associated to this controller in order to assess the limits of the GPC law and to suggest possible improvements. In addition, the GPC performance index takes into account the setpoint tracking problem, which is a natural approach to many industrial applications. State-space model formulations, such as those of (Rawlings and Muske, 1993; Scokaert and Mayne, 1998), usually tackle the problem of driving the state to the origin or "regulation problem", in which state feedback and thus state measurements or observers are required. This can become a relevant inconvenience, since usually only output measurements are available. Moreover, these methods require some manipulations to solve the setpoint tracking[1] problem. Finally, transfer function formulation are often preferred by the industrial community since most of the available system identification packages provide with transfer function models.

However, the choice between state-space or transfer function formulations is a matter of taste rather than reason. For instance, state-space descriptions are usually preferred for modelling MIMO systems, but transfer function MPC formulations can be used in the multi-variable case with no difficulty. It is worth pointing out that most of the results presented below can be extended to state-space MPC in a straightforward form, and thus the question of which representation should be used is not that relevant.

The discussion Chapters 2 through 5 include a **survey** of the state of the art of different MPC topics, from the basic formulations and definitions to robustness and stability issues, and then the **contributions** of this PhD research are reported and compared with the methodologies suggested in the literature.

The outline of this thesis is as follows:

Chapter 2 is devoted to the formulation of several "GPC-like" predictive controllers. Apart from the classical GPC of (Clarke *et al.*, 1987), the stabilising *Constrained Receding-Horizon Predictive Control* (CRHPC) and *Infinite horizon GPC* (GPC^{∞}) are **reviewed.** On the other hand, 1-norm cost functions are also taken

[1] The *Predictive Functional Control* (PFC) of Richalet (1993*b*) is a notable exception to this rule, since it solves the setpoint tracking problem using state-space models.

into account, leading to **1-norm counterparts of the standard 2-norm controllers**. The 1-norm GPC (GPC_1), 1-norm CRHPC ($CRHPC_1$) and 1-norm GPC^{∞} (GPC_1^{∞}) are thus formulated. However, it is noted that the GPC_1^{∞} solution requires an iterative algorithm, which can sometimes involve a large computational burden. To overcome this difficulty, **an upper bound of the GPC_1^{∞} cost function is obtained**, leading to the *Quasi-Infinite horizon 1-norm GPC* ($QGPC_1^{\infty}$). Although this controller does not preserve the stability guarantees, it provides with a computationally efficient solution which leads to appropriate performance for the vast majority of systems. The usefulness of 1-norm controllers is clarified in Chapter 4. The issue of stability when the true system to be controlled and the assumed model coincide is undertaken. The most relevant available stability results are presented, and **stability theorems for 1-norm controllers are also obtained**. The stability proofs, based on the monotonicity of the optimal cost function sequence, provide with an intuitive insight to the stability topic. The stabilising controllers are tested on several benchmark systems to show that the stability problems of the classical finite horizon GPC are overcome, and the computational simplicity of $QGPC_1^{\infty}$ compared to the GPC_1^{∞} is remarked. Finally, a **convergence property** from the $QGPC_1^{\infty}$ to the GPC_1^{∞} is conjectured.

Chapter 3 presents the classical approach to robustness for unconstrained 2-norm GPC-like controllers. This analysis is based on the equivalent *Linear Time Invariant* (LTI) formulation of GPC due to (Bitmead *et al.*, 1990). **A novel feature introduced in this chapter is the extension of this kind of analysis to the unconstrained infinite horizon GPC**, which is also formulated as an LTI controller. Robust stability conditions are provided for different uncertainty descriptions, *e.g.* additive and inverse multiplicative. Some of these descriptions make it possible the robustness analysis even when there is a changing number of unstable poles between the model and the true process. The aim of this chapter is twofold. On the one hand, **the performance and robustness properties of different stabilising controllers are analysed and compared.** This comparative study leads to the conclusion that the infinite horizon approach overcomes the equality constrained CRHPC as robustness is concerned, especially for typical choices of the tuning knobs. On the other hand, two classical methods to enhance the robustness of predictive controllers are reviewed, namely the T-design and the Q-parametrisation methods. The former is based on tuning the noise polynomial of the system model using heuristic rules, whereas the latter relies on parametrising via a rational function Q, all the controllers which lead to the same nominal transfer function, and then Q is chosen to maximise a robustness criterion. Throughout this analysis, it is observed that the effect of the polynomial T is similar for 1-norm and 2-norm controllers. Finally, **a new procedure to enhance the robustness of unconstrained GPC-like controllers, based on obtaining the polynomial T by means of optimising a robustness criterion instead of heuristic rules, is developed and compared to the existing approaches.** This procedure, termed as T-optimisation, is shown to overcome the heuristic T-design and the Q-parametrisation methods for a particular example.

Chapter 4 faces the robustness problem of constrained systems using MPC controllers. To begin with, different approaches to obtain robust constrained MPC, such as the methods of (Allwright, 1994; Camacho and Bordóns, 1995; Kothare *et al.*, 1996) are reviewed. The basis of all these control schemes is to compute the hypothesised control sequence to achieve the *minimum* of the *maximum* of a given cost function as the uncertainty ranges within some specified limits. This worst case approach is known as *min-max MPC.* These methods, however, use different types of models for the process and the uncertainty. **The global**

uncertainty description introduced in (Camacho and Bordóns, 1995) **is exploited in this chapter. This formulation is applied to the stabilising controllers described in Chapter 2, especially to (quasi) infinite horizon controllers** since these enjoy better robustness properties compared to the CRHPC. The global uncertainty is an unknown bounded signal which, added to the model output, produces the measured (or true) output. The min-max problem has been solved for both 2-norm and 1-norm controllers. In the former case, an analytical solution is shown to be untractable, and thus a numerical alternative based on non-linear programming is presented. On the other hand, the 1-norm case is solved as a simple LP problem. As efficiency is concerned, the 1-norm formulation is superior to the 2-norm counterpart, especially for certain tuning settings. In addition, the closed-loop behaviour of the uncertainty signal is investigated and **a band updating algorithm is suggested to modify on-line the assumed limits of the global uncertainty parameter.** This procedure, based on a few tuning knobs, is shown to provide with a convenient description of the uncertainty dynamics what makes it possible to replace the initial settings by less conservative counterparts. Tuning guidelines for the band updating strategy are also suggested. Several simulated examples are provided to illustrate the performance of these min-max controllers in front of modelling errors and disturbances. As illustrated with a few examples, the min-max MPC methods described in this chapter overcome the classical approach to robustness when constraints are considered. In addition, the min-max approach is tested on a strongly non-linear system, and it is observed that **some difficult non-linearities can be handled.** Various comparative studies of min-max MPC versus other control strategies are also provided. Finally, **a robustness analysis based on the statistical learning theory** (Vidyasagar, 1997) **is performed** to show that the min-max approach can often overcome the classical T-design method, especially as constraint handling is concerned.

Chapter 5 introduces some refinements of the min-max method based on the global uncertainty approach. This chapter is mainly focused on the *feedback formulation* of min-max MPC described by Scokaert and Mayne (1998). The key idea is to consider **different control profiles for different uncertainty realisations** together with an additional "causality constraint". This approach is intended to solve some of the problems related to the classical min-max controllers which are pointed out in Chapter 4. As discussed in (Scokaert and Mayne, 1998), a few of these difficulties stem from the use of a single control move profile to handle all the possible uncertainty realisations, ignoring the fact that feedback is present in the receding-horizon implementation of the controller. **The min-max GPC$^{\infty}$ is then adapted to exploit this possibility, and some simulated examples are provided to show that some of the drawbacks of the classical min-max methods can be avoided.** However, it is worth pointing out that the feedback min-max methods involve a larger computational burden compared to the classical min-max MPC. Finally a few directions to further the developments of min-max MPC are outlined.

Chapter 6 draws the conclusions from the results obtained throughout this PhD research and suggests possible directions for future research.

Appendix A describes some of the benchmark systems used throughout this thesis to test different control schemes.

Chapter 2

Formulation of Receding-Horizon Predictive Controllers

2.1 Introduction

The first few controllers in the MPC family, such as IDCOM, DMC and GPC, sometimes referred to as "first generation" MPC, use a finite prediction horizon, which is, at the same time, the reason for their early success and the cause for the criticism which arose later. This feature made it possible to incorporate constraints in a natural manner within the control strategy, a capability which is not supported by (infinite horizon) LQ control and, undoubtedly, constraint handling is one of the most appealing issues of MPC. However, there is also a major drawback common to all these early methods, since it is now widely accepted that no successful stability results exist for finite horizon formulations, as pointed out in (Bitmead *et al.*, 1990).

The stability flaws of the first generation MPC were overcome by a "second generation" of predictive controllers, which can be divided into two categories. The first one consists of methods such as the CRHPC (Clarke and Scattolini, 1991), the *Stabilising Input/Output Receding-Horizon Control* (SIORHC) (Mosca and Zhang, 1992), and the *Stable Generalised Predictive Control* (SGPC) (Kouvaritakis *et al.*, 1992), which enforce end-point equality constraints on the predicted outputs. These constraints are explicit in the CRHPC and the SIORHC, but implicit in the SGPC formulation. Although these three controllers have been proved to be theoretically equivalent, the latter enjoys better numerical properties, as remarked by Rossiter and Kouvaritakis (1994). Due to this equivalence, the term CRHPC is used hereafter to refer to either of these three approaches.

The second category is formed by algorithms which use an infinite prediction horizon, such as the state-space controller of (Rawlings and Muske, 1993), the infinite horizon GPC of (Scokaert, 1994; Scokaert, 1997), or the *Infinite Horizon Stable Predictive Control* (IHSPC) of (Rossiter *et al.*, 1996). These controllers can be exactly solved since they can be converted, after some manipulations, to an equivalent finite horizon problem with system dependent weighting matrices. Although infinite horizon predictive controllers might look like a re-discovery of LQ control, it must be remarked that they solve the difficulty of handling constraints within an infinite prediction horizon framework, using, for example, the methods described in (Rawlings and Muske, 1993).

In (Scokaert, 1994) a distinction is made between "second generation" and "third generation" MPC methods, to refer to the infinite horizon controllers and to the CRHPC "family" respectively. There are historical reasons for such a classification, since the first few implementations of infinite horizon controllers were only approximate, whereas the CRHPC can be exactly solved. However, newer developments have yielded an exact solution for infinite horizon predictive controllers, and thus such a distinction is not considered throughout this thesis.

The stabilising properties of the CRHPC family were early proved by showing the equivalence of these controllers with those proposed by Kleinman (1974) and Kwon and Pearson (1978). In other words, the first few proofs (Clarke and Scattolini, 1991; Mosca and Zhang, 1992) are based on the monotonicity of the co-variance matrix of the associated Riccati equations. More recent results (Kouvaritakis *et al.*, 1992; Scokaert, 1994) establish stability through the monotonicity of the optimal cost function values sequence, as previously done in (Mayne and Michalska, 1990) for continuous time receding-horizon control. The latter proofs are available for both the CRHPC and the infinite horizon MPC, and have contributed with an intuitive input/output domain approach to the issue of stability (Scokaert, 1994).

This chapter is concerned with the formulation of predictive controllers with input/output models, termed as GPC-like controllers henceforth. Both 2-norm and 1-norm formulations are presented, and stabilising methods are highlighted. The formulations are provided for the SISO case only for simplicity of notation, and extend to the MIMO framework in a straightforward manner. In addition, robustness features are not in the scope of this chapter, hence the nominal case (no modelling errors) is considered in the next sections. The minimisation of 1-norm cost functions tackled in this chapter is a necessary step towards the development of robust MPC controllers based on min-max optimisation which, as clarified in Chapter 4, can be solved much more efficiently if 1-norm cost functions are used instead of 2-norm counterparts. The 1-norm formulations introduced below are intended to provide controllers which satisfy *at least* nominal stability such that they need not be tuned to obtain stability. These nominally stabilising controllers are used as the basis of min-max robust controllers, as discussed in Chapter 4.

If 1-norm cost functions are used, the minimisation problem with an infinite prediction horizon requires the implementation of an iterative algorithm which solves two LP problems at each iteration. This solution can give rise to an enormous computational burden, since the number of iterations cannot be established *a priori*. This drawback can make this method impractical for real applications and discourages the application of the min-max approach with this infinite horizon controller. As an alternative, a very efficient *upper bound* solution of the infinite horizon problem can be computed with a single LP problem. This kind of solution is very convenient for min-max formulations, as shown in Chapter 4, but the nominal stability guarantees are not preserved. However, it is shown that nominal stability with this controller is easily obtained, as the upper bound and the true infinite horizon solutions seem to converge as the control horizon increases. Hence this 1-norm controller provides with a reasonable trade-off between nominal stability and computational requirements what makes it a likely candidate to develop an *efficient* min-max *robust* MPC method.

The outline of this chapter is as follows. Firstly, Section 2.2 summarises the contributions of several authors, providing an overview of existing controllers with 2-norm cost functions, and attention is driven to nominal stability issues. After that, Sections 2.3 through 2.6 are devoted to methods developed during this PhD research. To begin with, stabilising 1-norm controllers are formulated in Section 2.3, where stability theorems are proved for these new control schemes. In Section 2.4, the newly proposed 1-norm methods are compared to the existing 2-norm ones by means of simulated experiments. In Section 2.5, a quasi-infinite horizon 1-norm predictive controller is shown to converge to a true infinite horizon counterpart, making it possible to reduce the computational burden. Finally, concluding remarks are presented in Section 2.6.

2.2 2-norm cost functions

This section presents a reformulation of RHPC provided in (Yoon and Clarke, 1995*b*). This algorithm uses a *Controlled Auto-Regressive Integrated Moving Average* (CARIMA) model of the system to be controlled:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{T(q^{-1})}{\Delta}\xi(t), \tag{2.1}$$

where $u(t)$ is the input signal, $y(t)$ is the output signal, A, B and T are known polynomials in the delay operator q^{-1}:

$$\begin{aligned} q^{-1}B(q^{-1}) &= b_1q^{-1} + b_2q^{-2} + \cdots + b_{n_b}q^{-n_b}, \\ A(q^{-1}) &= 1 + a_1q^{-1} + \cdots + a_{n_a}q^{-n_a}, \\ T(q^{-1}) &= t_0 + t_1q^{-1} + \cdots + t_{n_t}q^{-n_t}, \end{aligned} \tag{2.2}$$

and $\xi(t)$ is a zero-mean, stochastic disturbance signal. Notice that $B(q^{-1})$ defined above is a polynomial of order $n_b - 1$, and hence the system is supposed to have $n_b - 1$ zeroes and n_a poles. In addition, the polynomials $A(q^{-1})$ and $B(q^{-1})$ are assumed to share no common unstable root, *i.e.* the system of eqn.2.1 is *stabilisable and detectable.*

Remark 2.1 Although the system of eqn.2.2 is a unit delayed plant, no difficulty arises in considering greater delays $d > 1$. Slight straightforward modifications are required in the sequel in that case but, in order to avoid a cumbersome notation, the unit delayed case is used throughout this thesis. □□□

An optimal control move $\Delta u(t) = \Delta u(t|t)$ is computed by minimising the cost function

$$\begin{aligned} J_2(t) = &\sum_{j=N_1}^{N_y-1} \mu(j)\,[w(t+j|t) - y(t+j|t)]^2 \\ &+ \sum_{j=N_y}^{N_2} \frac{\mu(N_y)}{\gamma}[w(t+N_y|t) - y(t+j|t)]^2 \\ &+ \sum_{j=1}^{N_u} \rho(j)\Delta u^2(t+j-1|t), \end{aligned} \tag{2.3}$$

with respect to $\Delta u(t+j|t)$, $j = 0, \ldots, N_u - 1$, and subject to $\Delta u(t+j|t) = 0$ for $j \geq N_u$, where N_u is called the control horizon. In addition, $y(t+j|t)$ are predictions of the output performed at time t, $w(t+j|t)$ are future values of the setpoint, which are known at time t or assumed to be equal to the current value $w(t|t)$, N_1 and N_2 are the lower and upper costing (or prediction) horizons, $\mu(j)$ and $\rho(j)$ are positive weighting sequences, and $0 \leq \gamma \leq 1$ is used to impose a heavier weighting on the predicted tracking errors from N_y to N_2. Moreover, $w(t+j|t)$ and $\mu(j)$ are assumed to be equal to $w(t+N_y|t)$ and $\mu(N_y)$, respectively, for $j > N_y$, because it can provide better closed-loop behaviour (Yoon and Clarke, 1995*b*).

Remark 2.2 It is worth pointing out that, throughout this thesis, all the experiments have been made as if no future information about the setpoint was available, or

$$w(t+j|t) = w(t|t),$$

for all $j \geq 0$. □□□

As pointed out in (Yoon and Clarke, 1995*b*), by introducing a few manipulations in the computation of the minimum of eqn.2.3, it becomes possible to choose $\gamma = 0$, which forces $y(t+j|t) = w(t+N_y|t)$, for $j = N_y, \ldots, N_2$. These are end-point equality constraints[1] on the internal model output.

The cost function of eqn.2.3 can lead to several classical controllers such as the GPC (Clarke *et al.*, 1987), the CRHPC (Clarke and Scattolini, 1991) or the infinite horizon GPC or GPC$^\infty$ (Scokaert, 1997). To obtain these controllers, the tuning knobs must be chosen as follows:

1. For the GPC: $\gamma = 1$. The second and the third terms in the cost function $J_2(t)$ in eqn.2.3 can be rearranged as one summation to provide the classical cost definition of (Clarke *et al.*, 1987):

$$J_2(t) = \sum_{j=N_1}^{N_2} \mu(j)\left[w(t+j|t) - y(t+j|t)\right]^2 + \sum_{j=1}^{N_u} \rho(j)\Delta u^2(t+j-1|t). \qquad (2.4)$$

Remark 2.3 Strictly speaking, the equivalence with eqn.2.3 holds only if $\mu(j) = \mu(N_y)$ and $w(t+j|t) = w(t+N_y|t)$ for all $j > N_y$. However, this situation is quite common since constant setpoints (no preprogrammed inputs) and constant weighting are the typical case. □□□

[1] Equality constrained schemes are referred to as "unconstrained" hereafter. The term "constrained" is only used for controllers which use hard inequality input, output or state constraints.

2. For the CRHPC: $N_1 = 1$, $N = N_u - 1$, $N_y = N + 1$, $N_2 = N + m$ and $\gamma = 0$:

$$J_2(t) = \sum_{j=1}^{N} \mu(j)\,[w(t+j|t) - y(t+j|t)]^2 + \sum_{j=1}^{N+1} \rho(j)\Delta u^2(t+j-1|t), \qquad (2.5)$$

subject to $y(t+N+j|t) = w(t+N+j|t)$ for $j = 1, 2, \ldots, m$. This controller enforces m equality constraints on the predicted outputs ($\gamma = 0$), and the parameter N defines both the control and the prediction horizons. These definitions must satisfy that $N_u = N+1 \geq m$ in order that there are at least enough degrees of freedom to satisfy the m end-point equality constraints. If a delay $d > 1$ is present, the definition $N = N_u - 2 + d$ must be used instead of $N = N_u - 1$ to ensure closed-loop stability (Scokaert, 1994).

3. For the GPC$^\infty$: $N_y = \infty$ and $\mu(j) = 1\,\forall j$:

$$J_2(t) = \sum_{j=N_1}^{\infty} [w(t+j|t) - y(t+j|t)]^2 + \sum_{j=1}^{N_u} \rho(j)\Delta u^2(t+j-1|t). \qquad (2.6)$$

In this case the optimisation is performed over an infinite horizon and the second term of $J_2(t)$ in eqn.2.3 is not involved. $N_1 = 1$ is used in (Scokaert, 1997), but this parameter does not affect the stabilising properties of the GPC$^\infty$ (Scokaert, 1994), which are preserved with any choice of the lower costing horizon.

The stabilising properties of the GPC are deeply analysed in (Bitmead *et al.*, 1990), where some examples are provided in which the stability of some plants is difficult to obtain by tuning the GPC. It is well known that the (finite-horizon) GPC does not guarantee the stability of the nominal closed-loop system. In order to obtain nominal stability, the GPC must be *tuned ad-hoc* and, in some cases, it can become quite a tough issue to find the appropriate values of the tuning knobs.

The first approach to overcome this problem was the CRHPC. In (Clarke and Scattolini, 1991) several choices of N and m which guarantee nominal stability for any given system are provided. Nevertheless, there are still a few difficulties related to the

CRHPC. To begin with, if short (prediction and control) horizons are used it tends to producing a suboptimal deadbeat-like behaviour of the closed-loop system leading to low robustness margins (Megías *et al.*, 1999*a*). This situation can be overcome by choosing longer horizons, but then numerical stability problems can arise as pointed out in (Rossiter and Kouvaritakis, 1994).

The SGPC of (Kouvaritakis *et al.*, 1992) is theoretically equivalent to the CRHPC, but it is numerically more robust. The approach taken in the SGPC is the use of the Youla-Kučera parametrisation to obtain a stable closed-loop system, and then apply the GPC law to the resulting stable system. As remarked in (Rossiter *et al.*, 1998), the use of an internal stabilising loop provides several numerical advantages. However, it is worth pointing out that the equality constraints are still implicitly present in the SGPC formulation. These constraints are somewhat artificial since they are not really satisfied due to the receding-horizon implementation of the controller. Hence a suboptimal solution is obtained, as the dimension of the decision space is reduced by the number of equality constraints. Moreover, to avoid the deadbeat-like behaviour, long horizons must be chosen and this can cause problems if hard constraints are also involved since, firstly, the dimension of the optimisation problem to be solved can become very large and, secondly, infeasibility can arise.

A newer family of nominal stabilising predictive controllers were developed by using an infinite prediction horizon, firstly for state-space models (Rawlings and Muske, 1993) and later for input/output model (transfer function) formulations (Scokaert, 1997). The GPC$^\infty$ overcomes the drawbacks of the CRHPC/SGPC, avoiding both the numerical difficulties and the deadbeat-like behaviour with small (prediction and control) horizons, and provides with better robustness properties as shown in (Megías *et al.*, 1999*a*). However, as the decision variables in the GPC$^\infty$ optimisation are future values of the control moves, there is still a minor disadvantage in these schemes, namely that the control horizon N_u must be finite.

The IHSPC of Rossiter *et al.* (1996) provides with an alternative implementation which allows the input trajectories to be infinite sequences. On the other hand, some more recent approaches make use of the Youla-Kučera parametrisation and the optimisation is performed not on closed-loop signals but on the elements of a state feedback vector (Fikar and Engell, 1997; Fikar *et al.*, 1999). These controllers, referred to as *Youla-Kučera Predictive Control* (YKPC) make it possible to optimise over infinite control and prediction horizons with finitely many unknowns. Nevertheless, this approach is difficult to use with the global uncertainty description which is introduced in the subsequent chapters and has not been analysed throughout this research.

The optimisation problem is slightly different for the finite (Yoon and Clarke, 1995*b*) and the infinite (Scokaert, 1997) horizon approaches, and the required formulae for these two situations are provided in the next two sections. In the sequel, the notations used by different authors are respected as much as possible, but it must be taken into account that some definitions vary between the finite and the infinite horizon cases.

2.2.1 The finite horizon case

This section summarises the standard procedure to compute an optimal control move vector for finite horizon 2-norm cost functions. The reformulation of receding-horizon predictive control provided by Yoon and Clarke (1995*b*) is used here.

To begin with, a few useful definitions are introduced. Let the control move vector $\boldsymbol{\Delta u}(t)$ be

$$\boldsymbol{\Delta u}(t) = \left[\begin{array}{cccc} \Delta u(t|t) & \Delta u(t+1|t) & \dots & \Delta u(t+N_u-1|t) \end{array}\right]^{\mathrm{T}}, \tag{2.7}$$

the setpoint vectors $\boldsymbol{w}_1(t)$ and $\boldsymbol{w}_2(t)$ be

$$\begin{aligned} \boldsymbol{w}_1(t) &= \left[\begin{array}{cccc} w(t+N_1|t) & w(t+N_1+1|t) & \dots & w(t+N_y-1|t) \end{array}\right]^{\mathrm{T}}, \\ \boldsymbol{w}_2(t) &= \left[\begin{array}{cccc} w(t+N_y|t) & w(t+N_y|t) & \dots & w(t+N_y|t) \end{array}\right]^{\mathrm{T}}, \end{aligned}$$

the output prediction vectors $\boldsymbol{y}_1(t)$ and $\boldsymbol{y}_2(t)$ be

$$\boldsymbol{y}_1(t) = \begin{bmatrix} y(t+N_1|t) & y(t+N_1+1|t) & \dots & y(t+N_y-1|t) \end{bmatrix}^{\mathrm{T}},$$
$$\boldsymbol{y}_2(t) = \begin{bmatrix} y(t+N_y|t) & y(t+N_y+1|t) & \dots & y(t+N_2|t) \end{bmatrix}^{\mathrm{T}},$$

and the *free response* vectors $\boldsymbol{f}_1(t)$ and $\boldsymbol{f}_2(t)$ be

$$\boldsymbol{f}_1(t) = \begin{bmatrix} f(t+N_1|t) & f(t+N_1+1|t) & \dots & f(t+N_y-1|t) \end{bmatrix}^{\mathrm{T}},$$
$$\boldsymbol{f}_2(t) = \begin{bmatrix} f(t+N_y|t) & f(t+N_y+1|t) & \dots & f(t+N_2|t) \end{bmatrix}^{\mathrm{T}},$$

where $f(t+j|t)$ are predictions of the output performed at time t taking all the future control moves to be zero. With these vector definitions, the output prediction vectors can be written as[2]

$$\boldsymbol{y}_1 = \boldsymbol{f}_1 + \boldsymbol{G}_1\Delta\boldsymbol{u},$$
$$\boldsymbol{y}_2 = \boldsymbol{f}_2 + \boldsymbol{G}_2\Delta\boldsymbol{u},$$

where $\boldsymbol{G}_1\Delta\boldsymbol{u}$ and $\boldsymbol{G}_2\Delta\boldsymbol{u}$ are vectors formed by the so-called the *forced response*, the dynamic matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ are given by

$$\boldsymbol{G}_1 = \begin{bmatrix} g_{N_1} & g_{N_1-1} & \cdots & 0 \\ g_{N_1+1} & g_{N_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_y-1} & g_{N_y-2} & \cdots & g_{N_y-N_u} \end{bmatrix},$$

$$\boldsymbol{G}_2 = \begin{bmatrix} g_{N_y} & g_{N_y-1} & \cdots & 0 \\ g_{N_y+1} & g_{N_y} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & \cdots & g_{N_2-N_u-1} \end{bmatrix},$$

and g_k are coefficients of the step response of $q^{-1}B/A$ with $g_k = 0$ for all $k \le 0$. Furthermore, if the matrices $\boldsymbol{M}$ and $\boldsymbol{R}$ are defined as

$$\boldsymbol{M} = \mathrm{diag}\left[\mu(N_1), \mu(N_1+1), \dots, \mu(N_y-1)\right],$$
$$\boldsymbol{R} = \mathrm{diag}\left[\rho(1), \rho(2), \dots, \rho(N_u)\right],$$

the cost function $J_2(t)$ can be arranged in vector notation to yield

$$J_2(t) = \left[\boldsymbol{w}_1 - \boldsymbol{f}_1 - \boldsymbol{G}_1\Delta\boldsymbol{u}\right]^{\mathrm{T}} \boldsymbol{M} \left[\boldsymbol{w}_1 - \boldsymbol{f}_1 - \boldsymbol{G}_1\Delta\boldsymbol{u}\right] + \frac{\mu(N_y)}{\gamma}\left[\boldsymbol{w}_2 - \boldsymbol{f}_2 - \boldsymbol{G}_2\Delta\boldsymbol{u}\right]^{\mathrm{T}}\left[\boldsymbol{w}_2 - \boldsymbol{f}_2 - \boldsymbol{G}_2\Delta\boldsymbol{u}\right] + \Delta\boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}\Delta\boldsymbol{u}. \quad (2.8)$$

[2]To simplify the notation, the time variable t is dropped wherever it is possible.

In the unconstrained case, a global minimiser of $J_2(t)$ is found deriving eqn.2.8 with respect to $\boldsymbol{\Delta u}$ and equating to zero, to obtain

$$\boldsymbol{\Delta u}^{\text{opt}} = \left[\boldsymbol{G}_1^{\text{T}}\boldsymbol{M}\boldsymbol{G}_1 + \frac{\mu(N_y)}{\gamma}\boldsymbol{G}_2^{\text{T}}\boldsymbol{G}_2 + \boldsymbol{R}\right]^{-1} \left[\boldsymbol{G}_1^{\text{T}}\boldsymbol{M}(\boldsymbol{w}_1 - \boldsymbol{f}_1) + \frac{\mu(N_y)}{\gamma}\boldsymbol{G}_2^{\text{T}}(\boldsymbol{w}_2 - \boldsymbol{f}_2)\right]. \quad (2.9)$$

Obviously, the solution provided in eqn.2.9 cannot be used when $\gamma = 0$. However, this small difficulty can be overcome by rearranging eqn.2.9 as

$$\left(\boldsymbol{G}_1^{\text{T}}\boldsymbol{M}\boldsymbol{G}_1 + \boldsymbol{R}\right)\boldsymbol{\Delta u}^{\text{opt}} + \boldsymbol{G}_2^{\text{T}}\boldsymbol{p}^{\text{opt}} = \boldsymbol{G}_1^{\text{T}}\boldsymbol{M}(\boldsymbol{w}_1 - \boldsymbol{f}_1), \quad (2.10)$$

where the vector $\boldsymbol{p}$ is defined such that

$$\boldsymbol{p}^{\text{opt}} = \frac{\mu(N_y)}{\gamma}\left[\boldsymbol{G}_2\boldsymbol{\Delta u}^{\text{opt}} - (\boldsymbol{w}_2 - \boldsymbol{f}_2)\right]. \quad (2.11)$$

Finally, the linear equation system provided by eqn.2.10 and 2.11 can be expressed as

$$\begin{bmatrix} \boldsymbol{G}_1^{\text{T}}\boldsymbol{M}\boldsymbol{G}_1 + \boldsymbol{R} & \boldsymbol{G}_2^{\text{T}} \\ \boldsymbol{G}_2 & -\dfrac{\gamma}{\mu(N_y)}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta u} \\ \boldsymbol{p} \end{bmatrix}^{\text{opt}} = \begin{bmatrix} \boldsymbol{G}_1^{\text{T}}\boldsymbol{M}(\boldsymbol{w}_1 - \boldsymbol{f}_1) \\ \boldsymbol{w}_2 - \boldsymbol{f}_2 \end{bmatrix}, \quad (2.12)$$

where $\boldsymbol{I}$ is the identity matrix of conformal dimension. The optimal control move vector $\boldsymbol{\Delta u}^{\text{opt}}$ can be obtained directly from eqn.2.12, or using the block matrix inversion lemma (Yoon and Clarke, 1995*b*), but the latter possibility is not advisable for implementation because of numerical considerations. When $\gamma = 0$, the vector $\boldsymbol{p}^{\text{opt}}$ turns out to be the Lagrangian *multipliers* associated to the equality constraints $\boldsymbol{G}_2\boldsymbol{\Delta u} = \boldsymbol{w}_2 - \boldsymbol{f}_2$.

On the other hand, a constrained minimum of $J_2(t)$ can be also obtained using standard QP methods. In that case, the minimisation is performed taking into account a set of general constraints of the form

$$\boldsymbol{P}\boldsymbol{\Delta u} \leq \boldsymbol{r}, \quad (2.13)$$

which can be used, as shown by Kuznetsov and Clarke (1996), to bound the input, the output, the input or output rates or accelerations, the internal states, etc. In general, it is possible to place constraints on any variable which can be written as a linear combination of the control moves and the free response. Finally, the constrained minimum is found as

$$\overline{\boldsymbol{\Delta u}^{\rm opt}} = \arg\min_{\boldsymbol{\Delta u}} J_2(t) \text{ subject to } \boldsymbol{P\Delta u} \leq \boldsymbol{r}.$$

Remark 2.4 When $\gamma = 0$ (if there are end-point equality constraints) the second term in the cost function J_2 of eqn.2.3 must be added to the QP problem as additional constraints: $\boldsymbol{G}_2\boldsymbol{\Delta u} = \boldsymbol{w}_2 - \boldsymbol{f}_2$. □□□

2.2.2 The infinite horizon case

This section provides a summary of the formulae proposed in (Scokaert, 1997) for the GPC$^\infty$. The future setpoints are assumed to be equal to the current value: $w(t+j|t) = w(t|t)$ for all $j \geq 0$, and, without loss of generality, the lower costing horizon N_1 is taken to be 1.

As done for the finite horizon case, a few definitions are introduced prior to undertake the minimisation of $J_2(t)$ (eqn.2.6). Let the prediction horizon[3] N be

$$N = \max\{N_u + n_b - 1, n_{\bar{a}}, n_t\}, \tag{2.14}$$

where $n_{\bar{a}}$ is the degree of the stable factor of A (see below). In addition, let $\boldsymbol{\Delta u}(t)$ be defined as in eqn.2.7, and $\boldsymbol{w}(t)$, $\boldsymbol{y}(t)$ and $\boldsymbol{f}(t)$ be

$$\begin{aligned} \boldsymbol{w}(t) &= \left[\ w(t+1|t) \quad w(t+2|t) \quad \ldots \quad w(t+N|t)\ \right]^{\rm T}, \\ \boldsymbol{y}(t) &= \left[\ y(t+1|t) \quad y(t+2|t) \quad \ldots \quad y(t+N|t)\ \right]^{\rm T}, \\ \boldsymbol{f}(t) &= \left[\ f(t+1|t) \quad f(t+2|t) \quad \ldots \quad f(t+N|t)\ \right]^{\rm T}. \end{aligned} \tag{2.15}$$

The vector $\boldsymbol{y}(t)$ of output predictions can be written as the sum of the free and forced responses:

$$\boldsymbol{y}(t) = \boldsymbol{f}(t) + \boldsymbol{G\Delta u}(t),$$

[3] The infinite horizon problem can be converted into a finite horizon one, as shown below.

where the dynamic matrix $\boldsymbol{G}$ is given by

$$\boldsymbol{G} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & \cdots & g_{N-N_u+1} \end{bmatrix}. \tag{2.16}$$

At this point, the predicted tracking error at time $t+j$ is defined as

$$e(t+j|t) = w(t+j|t) - y(t+j|t),$$

and a vector $\boldsymbol{e}(t)$ can be built as

$$\boldsymbol{e}(t) = \begin{bmatrix} e(t+1|t) & e(t+2|t) & \ldots & e(t+N|t) \end{bmatrix}^{\mathrm{T}}. \tag{2.17}$$

From the above definitions, it follows that

$$\begin{aligned} \boldsymbol{e}(t) &= \boldsymbol{w}(t) - \boldsymbol{y}(t) \\ &= \boldsymbol{w}(t) - \boldsymbol{f}(t) - \boldsymbol{G}\Delta\boldsymbol{u}(t). \end{aligned}$$

To proceed with the optimisation, it is convenient to split the system into its stable and unstable parts. The model denominator can be factorised as

$$A(q^{-1}) = \bar{A}(q^{-1})\widetilde{A}(q^{-1}),$$

where $\bar{A}(q^{-1})$ is a polynomial of strictly stable roots and $\widetilde{A}(q^{-1})$ is a polynomial of unstable roots. Furthermore, let the coefficients of these two polynomials be

$$\begin{aligned} \bar{A}(q^{-1}) &= 1 + \bar{a}_1 q^{-1} + \cdots + \bar{a}_{n_{\bar{a}}} q^{-n_{\bar{a}}}, \\ \widetilde{A}(q^{-1}) &= 1 + \tilde{a}_1 q^{-1} + \cdots + \tilde{a}_{n_{\tilde{a}}} q^{-n_{\tilde{a}}}. \end{aligned}$$

Hence, the unstable part of the output can be defined as

$$\begin{aligned} \tilde{y}(t) &= \frac{B(q^{-1})}{\widetilde{A}(q^{-1})} u(t-1) + \frac{T(q^{-1})}{\widetilde{A}(q^{-1})\Delta} \xi(t) \\ &= \bar{A}(q^{-1}) y(t). \end{aligned} \tag{2.18}$$

Now, if the dynamic matrix $\widetilde{\boldsymbol{G}}$ is provided by

$$\widetilde{\boldsymbol{G}} = \begin{bmatrix} \tilde{g}_N & \tilde{g}_{N-1} & \cdots & \tilde{g}_{N-N_u+1} \\ \tilde{g}_{N+1} & \tilde{g}_N & \cdots & \tilde{g}_{N-N_u+2} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{g}_{N+n_{\tilde{a}}} & \tilde{g}_{N+n_{\tilde{a}}-1} & \cdots & \tilde{g}_{N+n_{\tilde{a}}-N_u+1} \end{bmatrix},$$

where $\tilde{g}_k$ is the k^{th} step response coefficient of $q^{-1}B(q^{-1})/\widetilde{A}(q^{-1})$, then it is possible to define vectors of predictions for the unstable part of output:

$$\begin{aligned}\widetilde{\boldsymbol{f}}(t) &= \left[\ \tilde{f}(t+N|t) \quad \tilde{f}(t+N+1|t) \quad \dots \quad \tilde{f}(t+N+n_{\tilde{a}}|t)\ \right]^{\mathrm{T}},\\ \widetilde{\boldsymbol{y}}(t) &= \left[\ \tilde{y}(t+N|t) \quad \tilde{y}(t+N+1|t) \quad \dots \quad \tilde{y}(t+N+n_{\tilde{a}}|t)\ \right]^{\mathrm{T}},\end{aligned} \tag{2.19}$$

where $\tilde{f}(t+j|t)$ denotes the free response of the system of eqn.2.18. Finally, the vector of future predictions can be computed as

$$\widetilde{\boldsymbol{y}}(t) = \widetilde{\boldsymbol{f}}(t) + \widetilde{\boldsymbol{G}}\Delta\boldsymbol{u}(t).$$

In order to write the cost function in a form which makes the optimisation task easier, a set of end-point equality constraints on the unstable part of the output are taken into account:

$$\tilde{y}(t+N+j) = \bar{A}(1)w(t+N+j|t),\ \forall j = 0, 1, \dots, n_{\tilde{a}}, \tag{2.20}$$

or, in vector form,

$$\widetilde{\boldsymbol{y}}(t) = \widetilde{\boldsymbol{w}}(t),$$

with

$$\widetilde{\boldsymbol{w}}(t) = \bar{A}(1)\left[\ w(t+N|t) \quad w(t+N+1|t) \quad \dots \quad w(t+N+n_{\tilde{a}}|t)\ \right]^{\mathrm{T}}. \tag{2.21}$$

As highlighted in (Scokaert, 1997), these constraints are redundant and do not modify the GPC$^{\infty}$ control law for $N_u > n_{\tilde{a}}$ (which is required for stability), since violation eqn.2.20 would result in an unbounded (and thus not minimal) cost. However, the end-point constraints make it much simpler to proceed with the minimisation.

Remark 2.5 The end-point constraints of eqn.2.20 hold not only for $j = 0, 1, \dots, n_{\tilde{a}}$, but for all $j \geq 0$, due to the definition of N. Therefore, all the unstable modes of the system are zeroed by time $t + N + n_{\tilde{a}}$. ◻◻◻

Now, eqn.2.20 can be rearranged as

$$\bar{A}(q^{-1})y(t+N+j|t) = \bar{A}(1)w(t+N+j|t),\ \forall j \geq 0,$$

and, if $\bar{A}(q^{-1})w(t+N+j|t)$ is subtracted from both sides, it follows that

$$\bar{A}(q^{-1})e(t+N+j|t) = \bar{A}(1)w(t+N+j|t) - \bar{A}(q^{-1})w(t+N+j|t). \qquad (2.22)$$

Thus, since $N \geq n_{\bar{a}}$ and $w(t+N+j|t) = w(t|t)$ for all $N+j \geq 0$, eqn.2.22 becomes

$$\bar{A}(q^{-1})e(t+N+j|t) = 0. \qquad (2.23)$$

Therefore, the predicted tracking errors decay exponentially to zero after time $t+N$ according to the dynamics of the stable modes of the system, the series converges, and thus it is possible to compute the infinite sum. Notice also that the control law is identical for all $N_1 \geq N$, since the predicted tracking errors from $t+N$ on evolve as per eqn.2.23 whatever value N_1 takes.

First of all, let the cost function be rewritten in a more convenient form:

$$J_2(t) = \sum_{j=1}^{N-1} e(t+j|t)^2 + \sum_{j=0}^{\infty} e(t+N+j|t)^2 + \sum_{j=1}^{N_u} \rho(j)\Delta u(t+j-1|t)^2.$$

The second term of this expression can be exactly computed taking into account the comments reported above. Now, let the matrices $\boldsymbol{R}$ and $\boldsymbol{Q}$ be defined as

$$\boldsymbol{R} = \mathrm{diag}\left[\rho(1), \rho(2), \ldots, \rho(N_u)\right],$$
$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{I}_{N-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix},$$

where $\boldsymbol{I}_k$ is the identity matrix of dimension k and $\boldsymbol{0}$ is the zero matrix of appropriate dimension. Notice that for $N_1 > 1$ the first N_1 diagonal coefficients of $\boldsymbol{I}_{N-1}$ in $\boldsymbol{Q}$ should be replaced by 0. In addition, $\boldsymbol{Q}$ is $\boldsymbol{0}$ for $N_1 \geq N$.

Using these definitions, the cost function becomes

$$J_2(t) = \boldsymbol{e}(t)^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{e}(t) + \boldsymbol{\Delta u}(t)^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Delta u}(t) + \sum_{j=0}^{\infty} e(t+N+j|t)^2, \qquad (2.24)$$

where only the last term needs further manipulations. A state-space realisation of the stable part of the model is used to solve the series. To begin with, the vector

$$z(t) = \begin{bmatrix} e(t+N-n_{\bar{a}}+1|t) & e(t+N-n_{\bar{a}}+2|t) & \dots & e(t+N|t) \end{bmatrix}^{\mathrm{T}} \tag{2.25}$$

and the matrix

$$\Phi = \begin{bmatrix} \mathbf{0}_{n_{\bar{a}}-1,1} & & I_{n_{\bar{a}}-1} & \\ -\bar{a}_{n_{\bar{a}}} & -\bar{a}_{n_{\bar{a}}-1} & \dots & -\bar{a}_1 \end{bmatrix} \tag{2.26}$$

are introduced, and $\mathbf{0}_{k,l}$ denotes the zero matrix of dimension $k \times l$. The equality constraints of eqn.2.23 imply that

$$z(t+j) = \Phi z(t+j-1),$$

or

$$z(t+j) = \Phi^j z(t). \tag{2.27}$$

Now let C be the row vector

$$C = \begin{bmatrix} 0 & \overset{(n_{\bar{a}}-1)}{\dots} & 0 & 1 \end{bmatrix}, \tag{2.28}$$

hence

$$e(t+N+j|t) = Cz(t+j) = C\Phi^j z(t),$$

which leads to

$$\begin{aligned} \sum_{j=0}^{\infty} e(t+N+j|t)^2 &= \sum_{j=0}^{\infty} z(t)^{\mathrm{T}} \Phi^{j^{\mathrm{T}}} C^{\mathrm{T}} C \Phi^j z(t) \\ &= z(t)^{\mathrm{T}} \sum_{j=0}^{\infty} \left(\Phi^{j^{\mathrm{T}}} C^{\mathrm{T}} C \Phi^j \right) z(t). \end{aligned}$$

Now, if $\overline{Q}$ is defined as

$$\overline{Q} = \sum_{j=0}^{\infty} \Phi^{j^{\mathrm{T}}} C^{\mathrm{T}} C \Phi^j,$$

it follows that $\overline{Q}$ satisfies the matrix Lyapunov equation (Scokaert, 1997)

$$\overline{Q} = C^{\mathrm{T}} C + \Phi^{\mathrm{T}} \overline{Q} \Phi,$$

which can be solved using existing standard tools.

Remark 2.6 Some formulations of infinite horizon predictive controllers have been proposed in such a way that the need for solving the Lyapunov equation is avoided (Rossiter *et al.*, 1996). □□□

Once $\overline{\boldsymbol{Q}}$ is available, it is possible to write

$$\sum_{j=0}^{\infty} e(t+N+j|t)^2 = z(t)^{\mathrm{T}}\overline{\boldsymbol{Q}}z(t) = \boldsymbol{e}(t)^{\mathrm{T}}\boldsymbol{Q}'\boldsymbol{e}(t),$$

for

$$\boldsymbol{Q}' = \begin{bmatrix} \boldsymbol{0}_{N-n_{\tilde{a}}} & \boldsymbol{0} \\ \boldsymbol{0} & \overline{\boldsymbol{Q}} \end{bmatrix}.$$

And finally, if $\boldsymbol{\Lambda}$ is defined as

$$\boldsymbol{\Lambda} = \boldsymbol{Q} + \boldsymbol{Q}',$$

it follows from eqn.2.24 that

$$J_2(t) = \boldsymbol{e}(t)^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{e}(t) + \boldsymbol{\Delta u}(t)^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Delta u}(t).$$

which is a finite dimensional quadratic cost function. Surprisingly enough, the infinite horizon problem turns out to be equivalent to a finite dimensional one if system dependent weighting ($\overline{\boldsymbol{Q}}$) is used and the end-point constraints of eqn.2.20 are enforced.

In the unconstrained case, the optimal control move vector is found at

$$\boldsymbol{\Delta u}^{\mathrm{opt}} = \arg\min_{\boldsymbol{\Delta u}} J_2(t) \text{ subject to } \widetilde{\boldsymbol{y}} = \widetilde{\boldsymbol{w}},$$

where the equality constraints on the unstable part of the output can be loosely considered as an infinity weighted term in the cost function:

$$J_2(t) = [\boldsymbol{w} - \boldsymbol{f} - \boldsymbol{G}\boldsymbol{\Delta u}]^{\mathrm{T}}\boldsymbol{\Lambda}[\boldsymbol{w} - \boldsymbol{f} - \boldsymbol{G}\boldsymbol{\Delta u}] + \infty\left[\widetilde{\boldsymbol{w}} - \widetilde{\boldsymbol{f}} - \widetilde{\boldsymbol{G}}\boldsymbol{\Delta u}\right]^{\mathrm{T}}\left[\widetilde{\boldsymbol{w}} - \widetilde{\boldsymbol{f}} - \widetilde{\boldsymbol{G}}\boldsymbol{\Delta u}\right] + \boldsymbol{\Delta u}^{\mathrm{T}}\boldsymbol{R}\boldsymbol{\Delta u},$$

which is completely analogous to eqn.2.8 when $\gamma = 0$. Thus, the optimal control move vector and the Lagrangian *multipliers* can be obtained from

$$\begin{bmatrix} G^{\mathrm{T}}\Lambda G + R & \widetilde{G}^{\mathrm{T}} \\ \widetilde{G} & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \lambda \end{bmatrix}^{\mathrm{opt}} = \begin{bmatrix} G^{\mathrm{T}}\Lambda(w - f) \\ \widetilde{w} - \widetilde{f} \end{bmatrix}. \tag{2.29}$$

In the constrained case, general constraints in the form of eqn.2.13 are taken into account, and the optimisation problem can be posed as

$$\overline{\Delta u^{\mathrm{opt}}} = \arg\min_{\Delta u} J_2(t) \text{ subject to } \begin{cases} \widetilde{G}\Delta u = \widetilde{w} - \widetilde{f}, \\ P\Delta u \leq r. \end{cases}$$

2.2.3 Relevant stability theorems

Stability theorems for the CRHPC and GPC$^{\infty}$ are proved in (Clarke and Scattolini, 1991; Scokaert, 1994; Yoon, 1994; Scokaert, 1997). This section includes two of them for the sake of completeness.

Theorem 2.1 (Scokaert, 1994; Chischi and Mosca, 1994) *For any stabilisable and detectable system, the CRHPC leads to a stable closed-loop system if*

(i) $\mu(N) \geq \mu(N-1) \geq \cdots \geq \mu(1)$,

(ii) $\rho(N_u) \geq \rho(N_u - 1) \geq \cdots \geq \rho(1) \geq 0$, *and*

(iii) $m = \max\{n_a, n_b - 1\} + 1$.

Remark 2.7 $N_u = N+1$ must be chosen to be greater than or equal to m (the number of equality constraints). ❑❑❑

Remark 2.8 As the number of end-point equality constraints (m) is greater than the system orders, these constraints are satisfied not only for $t+N+1, t+N+2, \ldots, t+N+m$, but for all $t+N+j$ with $j > 0$. ❑❑❑

Remark 2.9 A weaker statement is proved in (Clarke and Scattolini, 1991) by showing the equivalence between the CRHPC and the controller defined by Kwon and Pearson (1978). □□□

Remark 2.10 The proof given below was provided by Scokaert (1994), and is based on the monotonicity of the cost function (Lyapunov theory). □□□

Proof: Without loss of generality it can be assumed, for the sake of simplicity, that $\mu(j) = \bar{\mu}$ for all $1 \leq j \leq N$ and $\rho(j) = \bar{\rho}$ for all $1 \leq j \leq N_u$. As a consequence of Remark 2.8 the control move vector

$$\boldsymbol{\Delta u}^*(t+1) = \left[\; \Delta u^{\text{opt}}(t+1|t) \; \ldots \; \Delta u^{\text{opt}}(t+N_u-1|t) \;\; 0 \; \right]^{\text{T}}, \tag{2.30}$$

is feasible at time $t+1$. Let $J_2^{\text{opt}}(t)$ be the optimal cost at time t and $J_2^*(t+1)$ be the cost at time $t+1$ if the control move vector $\boldsymbol{\Delta u}^*(t+1)$ were implemented. From the cost function definition of the CRHPC (eqn.2.5) it follows that

$$\begin{aligned} J_2^{\text{opt}}(t) &= \bar{\mu} \sum_{j=1}^{N} e^2(t+j|t) + \bar{\rho} \sum_{j=1}^{N_u} \Delta u^2(t+j-1|t), \\ J_2^*(t+1) &= \bar{\mu} \sum_{j=2}^{N+1} e^2(t+j|t) + \bar{\rho} \sum_{j=2}^{N_u+1} \Delta u^2(t+j-1|t), \end{aligned}$$

for $N_u = N+1$, but $\Delta u(t+N_u|t) = 0$ and $e(t+N+1|t) = 0$ (Remark 2.8), thus

$$J_2^*(t+1) = \bar{\mu} \sum_{j=2}^{N} e^2(t+j|t) + \bar{\rho} \sum_{j=2}^{N_u} \Delta u^2(t+j-1|t),$$

hence

$$J_2^*(t+1) - J_2^{\text{opt}}(t) = -\bar{\mu} e^2(t+1|t) - \bar{\rho} \Delta u^2(t|t) \leq 0. \tag{2.31}$$

Therefore $J_2^*(t+1) \leq J_2^{\text{opt}}(t)$. In addition, $J_2^*(t+1)$ is only suboptimal at time $t+1$, and thus the optimal value of the cost function attained with $\boldsymbol{\Delta u}^{\text{opt}}(t+1)$ must satisfy $J_2^{\text{opt}}(t+1) \leq J_2^*(t+1)$. Now, combining these two inequalities:

$$0 \leq J_2^{\text{opt}}(t+1) \leq J_2^*(t+1) \leq J_2^{\text{opt}}(t), \tag{2.32}$$

that is, the sequence of optimal cost function values is non-increasing, and it is bounded below by zero, then it converges to some limit, say l. Making use of of eqn.2.31 and 2.32, it is possible to write

$$\begin{aligned} J_2^{\text{opt}}(t+1) - J_2^{\text{opt}}(t) &\leq J_2^*(t+1) - J_2^{\text{opt}}(t) \\ &= -\bar{\mu}e^2(t+1|t) - \bar{\rho}\Delta u^2(t|t), \end{aligned}$$

thus

$$\bar{\mu}e^2(t+1|t) + \bar{\rho}\Delta u^2(t|t) \leq J_2^{\text{opt}}(t) - J_2^{\text{opt}}(t+1), \tag{2.33}$$

Now, as $\{J_2^{\text{opt}}(t)\}$ converges to l, the right-hand side of eqn.2.33 decays to zero, and thus

$$\begin{cases} \lim_{t\to\infty} \bar{\mu}e(t+1) = \lim_{t\to\infty} \bar{\mu}e(t+1|t) = 0, \\ \lim_{t\to\infty} \bar{\rho}\Delta u(t) = \lim_{t\to\infty} \bar{\rho}\Delta u(t|t) = 0, \end{cases}$$

which implies stability for $\bar{\mu} \neq 0$ and $\bar{\rho} \neq 0$. ∇∇∇

Remark 2.11 This proof establishes that $J_2^{\text{opt}}(t)$ is a Lyapunov (non-increasing and bounded below by zero) function of the closed-loop system. ❑❑❑

Remark 2.12 The proof easily extends to the case of non-constant weighting, as shown in (Scokaert, 1994). ❑❑❑

Remark 2.13 Stability for $\bar{\mu} = 0$ and $\bar{\rho} = 0$ is proved in (Scokaert, 1994) using an optimality argument. However, notice that it is not possible to choose *both* ρ and μ to be zero *at the same time.* ❑❑❑

Remark 2.14 The proof extends to the constrained case only if the constraint horizon is infinity. However, there are some cases for which the stability guarantees are preserved with a finite constraint horizon. See (Scokaert, 1994) for details. ❑❑❑

Theorem 2.2 (Scokaert, 1994; Scokaert, 1997) *For any stabilisable and detectable system, the* GPC$^\infty$ *is stabilising if*

(i) $\rho(N_u) \geq \rho(N_u - 1) \geq \cdots \geq \rho(1) \geq 0$, *and*

(ii) $N_u > n_{\tilde{a}}$.

Remark 2.15 The second condition is necessary to allow the required degrees of freedom to enforce the end-point equality constraints on the unstable part of the model.

□□□

Remark 2.16 The proof provided below closely parallels that of Theorem 2.1 and was given by Scokaert (1994). □□□

Proof: Again, it can be assumed, without loss of generality, that the weighting sequence $\rho(j) = \bar{\rho}$ is constant, and let $\mathbf{\Delta u}^*(t+1)$ be defined as in eqn.2.30. Following an argument completely analogous to that of Remark 2.8, $\mathbf{\Delta u}^*(t+1)$ is feasible at time $t+1$, and then

$$J_2^{\text{opt}}(t) = \sum_{j=N_1}^{\infty} e^2(t+j|t) + \bar{\rho} \sum_{j=1}^{N_u} \Delta u^2(t+j-1|t),$$
$$J_2^*(t+1) = \sum_{j=N_1+1}^{\infty} e^2(t+j|t) + \bar{\rho} \sum_{j=2}^{N_u+1} \Delta u^2(t+j-1|t),$$

now, going through the same steps as in the proof of Theorem 2.1, it follows that

$$\begin{cases} \lim_{t\to\infty} e(t+N_1|t) = 0, \\ \lim_{t\to\infty} \bar{\rho}\Delta u(t) = \lim_{t\to\infty} \bar{\rho}\Delta u(t|t) = 0, \end{cases}$$

which implies stability for $\bar{\rho} \neq 0$. ▽▽▽

Remark 2.17 The proof easily extends to the case of a non-constant weighting in $\rho(j)$ (Scokaert, 1994). □□□

Remark 2.18 Stability for $\bar{\rho} = 0$ is proved in (Scokaert, 1994) using an optimality argument. □□□

Remark 2.19 The proof extends to the constrained case only if the constraint horizon is infinity. Without this condition the control profile postulated at time t does not have to be feasible at time $t+1$, and the monotonicity of the cost function cannot be ensured. Infinite horizon constraints can be implemented as described by Rawlings and Muske (1993). □□□

2.3 1-norm cost functions

As remarked in Section 2.1, in order to develop efficient robust predictive controllers with a global uncertainty approach it is convenient to use a 1-norm cost function instead as a 2-norm counterpart (Camacho and Bordóns, 1995), that is, to replace the "square" by the "absolute value" in the cost function definition:

$$\begin{aligned} J_1(t) = & \sum_{j=N_1}^{N_y-1} \mu(j)\,|w(t+j|t) - y(t+j|t)| \\ & + \sum_{j=N_y}^{N_2} \frac{\mu(N_y)}{\gamma}\,|w(t+N_y|t) - y(t+j|t)| \\ & + \sum_{j=1}^{N_u} \rho(j)\,|\Delta u(t+j-1|t)|\,. \end{aligned} \tag{2.34}$$

It is possible to define counterparts of the 2-norm GPC, CRHPC and GPC^∞, referred to as GPC_1, CRHPC_1 and GPC_1^∞ respectively hereafter, using the same tuning knobs as described in Section 2.2. 1-norm controllers can be implemented by solving simple LP problems (Camacho and Bordóns, 1995), for which very efficient standard solutions exist. However, in the 1-norm case, it is not as easy to manage the case $\gamma \to 0$ (or $\gamma = 0$) as done for the 2-norm cost functions, even in the absence of general inequality constraints $\boldsymbol{P}\boldsymbol{\Delta u} \leq \boldsymbol{r}$.

For $\gamma \neq 0$, the resulting controller is just a GPC_1 with a j-dependent weighting in the tracking errors. The formulae of the (finite horizon) GPC_1 are exhaustively documented in (Camacho and Bordóns, 1995), and thus are not repeated here for brevity. On the other hand, the rest of this section is devoted to the implementation aspects of the 1-norm version of the stabilising CRHPC and GPC$^\infty$. These require a few careful considerations, related to the end-point equality constraints and the infinite horizon, in the posing of the associated LP problems.

2.3.1 1-norm CRHPC

First of all, let the different symbols be defined as in Section 2.2.1, with $N_1 = 1$, $N = N_u - 1$, $N_y = N + 1$, $N_2 = N + m$ and $\gamma = 0$. The (constrained) CRHPC$_1$ is implemented on-line as the solution of

$$\overline{\Delta \boldsymbol{u}^{\mathrm{opt}}}(t) = \arg\min_{\Delta \boldsymbol{u}} J_1(t) \text{ subject to } \begin{cases} G_2 \Delta \boldsymbol{u} = \boldsymbol{w}_2 - \boldsymbol{f}_2, \\ \boldsymbol{P} \Delta \boldsymbol{u} \leq \boldsymbol{r}, \end{cases}$$

with

$$J_1(t) = \sum_{j=1}^{N} \mu(j) \left| e(t+j|t) \right| + \sum_{j=1}^{N_u} \rho(j) \left| \Delta u(t+j-1|t) \right|.$$

Now define additional variables $\sigma(j) \geq 0$ and $\beta(j) \geq 0$ such that

$$\begin{array}{rcccl} -\sigma(j) & \leq & e(t+j|t) & \leq & \sigma(j), \quad 1 \leq j \leq N, \\ -\beta(j) & \leq & \Delta u(t+j-1|t) & \leq & \beta(j), \quad 1 \leq j \leq N_u, \\ 0 & \leq & \sum_{j=1}^{N} \mu(j)\sigma(j) + \sum_{j=1}^{N_u} \rho(j)\beta(j) & \leq & \Psi, \end{array}$$

and the problem of minimising J_1 is equivalent to that of minimising the upper bound[4] Ψ.

Hence the optimisation can be performed as the LP problem[5]

$$\min_{\Psi, \boldsymbol{\sigma}, \boldsymbol{\beta}, \Delta \boldsymbol{u}} \Psi \text{ subject to } \begin{cases} \boldsymbol{\sigma} \geq G_1 \Delta \boldsymbol{u} + \boldsymbol{f}_1 - \boldsymbol{w}_1, \\ \boldsymbol{\sigma} \geq -G_1 \Delta \boldsymbol{u} - \boldsymbol{f}_1 + \boldsymbol{w}_1, \\ \boldsymbol{\beta} \geq \Delta \boldsymbol{u}, \\ \boldsymbol{\beta} \geq -\Delta \boldsymbol{u}, \\ \Psi \geq \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{\sigma} + \boldsymbol{\rho}^{\mathrm{T}} \boldsymbol{\beta}, \\ \boldsymbol{P} \Delta \boldsymbol{u} \leq \boldsymbol{r}, \\ G_2 \Delta \boldsymbol{u} = \boldsymbol{w}_2 - \boldsymbol{f}_2, \\ \boldsymbol{\sigma} \geq \boldsymbol{0}, \; \boldsymbol{\beta} \geq \boldsymbol{0}, \; \Psi \geq 0, \end{cases}$$

[4]As $\mu(j)$ and $\rho(j)$ are non-negative for all j, the minimum Ψ^{opt} is obtained for $\sigma^{\mathrm{opt}}(j) = |e(t+j|t)|$, $\beta^{\mathrm{opt}}(j) = |\Delta u(t+j|t)|$, and $\Psi^{\mathrm{opt}} = J_1^{\mathrm{opt}}(t)$.

[5]The vector inequalities of the form $\boldsymbol{u} \leq \boldsymbol{v}$ denote the componentwise inequalities $\boldsymbol{u}_j \leq \boldsymbol{v}_j$ for all j.

with

$$\begin{aligned}
\boldsymbol{\mu} &= [\ \mu(1)\ \ \mu(2)\ \ \ldots\ \ \mu(N)\ ]^{\mathrm{T}}, \\
\boldsymbol{\sigma} &= [\ \sigma(1)\ \ \sigma(2)\ \ \ldots\ \ \sigma(N)\ ]^{\mathrm{T}}, \\
\boldsymbol{\rho} &= [\ \rho(1)\ \ \rho(2)\ \ \ldots\ \ \rho(N_u)\ ]^{\mathrm{T}}, \\
\boldsymbol{\beta} &= [\ \beta(1)\ \ \beta(2)\ \ \ldots\ \ \beta(N_u)\ ]^{\mathrm{T}}.
\end{aligned} \tag{2.35}$$

Finally, this LP problem can be written in the standard form

$$\min_{\boldsymbol{x}} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \text{ subject to } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{\sigma} \geq \mathbf{0},\ \boldsymbol{\beta} \geq \mathbf{0},\ \Psi \geq 0,$$

with

$$\boldsymbol{x} = \left[\begin{array}{c} \Delta\boldsymbol{u} \\ \hline \boldsymbol{\sigma} \\ \hline \boldsymbol{\beta} \\ \hline \Psi \end{array}\right], \quad \boldsymbol{c} = \left[\begin{array}{c} \mathbf{0} \\ \hline \mathbf{0} \\ \hline \mathbf{0} \\ \hline 1 \end{array}\right],$$

and

$$\boldsymbol{A} = \left[\begin{array}{c|c|c|c} \boldsymbol{G}_1 & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{G}_1 & -\boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \hline \boldsymbol{I} & \mathbf{0} & -\boldsymbol{I} & \mathbf{0} \\ -\boldsymbol{I} & \mathbf{0} & -\boldsymbol{I} & \mathbf{0} \\ \hline \mathbf{0}^{\mathrm{T}} & \boldsymbol{\mu}^{\mathrm{T}} & \boldsymbol{\rho}^{\mathrm{T}} & -1 \\ \hline \boldsymbol{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hline \boldsymbol{G}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\boldsymbol{G}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right], \quad \boldsymbol{b} = \left[\begin{array}{c} -\boldsymbol{f}_1 + \boldsymbol{w}_1 \\ \boldsymbol{f}_1 - \boldsymbol{w}_1 \\ \hline \mathbf{0} \\ \mathbf{0} \\ \hline 0 \\ \hline \boldsymbol{r} \\ \hline -\boldsymbol{f}_2 + \boldsymbol{w}_2 \\ \boldsymbol{f}_2 - \boldsymbol{w}_2 \end{array}\right].$$

2.3.2 1-norm GPC$^\infty$

Now an infinite horizon cost function of the form

$$J_1(t) = \sum_{j=1}^{\infty} |e(t+j|t)| + \sum_{j=1}^{N_u} \rho(j)\,|\Delta u(t+j-1|t)|, \tag{2.36}$$

is taken into account[6]. Although N_1 has been assumed to be 1, any other value might be specified with minor modifications. The optimisation of this cost function would be, in

[6]Throughout this section the definitions of Section 2.2.2 are supposed to hold unless explicitly otherwise specified.

principle, solvable as an *infinite-dimensional LP problem*, that is to say, the formulation of Section 2.3.1 could be used taking the vector $\boldsymbol{\sigma}$ to be infinite-dimensional, taking $\boldsymbol{\mu}$ to be an infinite-dimensional ones vector, and removing the equality constraints $\boldsymbol{G}_2\Delta\boldsymbol{u} = \boldsymbol{w}_2 - \boldsymbol{f}_2$. However, it is not possible to solve such a problem exactly, since infinite-dimensional vectors cannot be handled by numerical algorithms. A possible approach is to look for approximate suboptimal solutions which converge to the optimal one. From an infinite-dimensional LP problem, it is possible to obtain upper bound and lower bound solutions using the *Finitely Many Variables* (FMV) and the *Finitely Many Equations* (FME) methods respectively (Staffans, 1993; Dahleh and Díaz-Bovillo, 1995). These converge to the solution of the truly infinite-dimensional problem, and thus provide with a way to get as close to the optimum as wished. It is only necessary to reduce the difference between the upper and the lower bounds as required.

Although the FMV and the FMV methods could be easily applied to the problem of minimising eqn.2.36, there are simpler ways of obtaining lower and upper bounds to $J_1(t)$ in this case. First of all, let the cost function be written as a three-term sum:

$$J_1(t) = \sum_{j=1}^{N-1} |e(t+j|t)| + \sum_{j=0}^{\infty} |e(t+N+j|t)| + \sum_{j=1}^{N_u} \rho(j)\,|\Delta u(t+j-1|t)|, \qquad (2.37)$$

with the prediction horizon N as defined in eqn.2.14. The end-point equality constraints of eqn.2.22 are considered in the minimisation of the cost function for computation purposes, since any solution to the infinite horizon problem must necessarily satisfy them, as remarked in (Scokaert, 1997). To simplify the notation below, let S_i^k be defined as

$$S_i^k = \begin{cases} \sum_{j=i}^{k} |e(t+N+j|t)|, & \text{if } k \geq i, \\ 0, & \text{otherwise,} \end{cases}$$

i.e., S_i^k the sum of the absolute value of the errors from $t+N+i$ to $t+N+k$ as predicted from information available at time t. Notice that the second term of eqn.2.37

can be denoted as S_0^∞. Now, the series S_0^∞ can then be split as

$$S_0^\infty = S_0^{n-1} + S_n^\infty,\ \forall n \geq 0.$$

An upper bound can be computed for the term S_n^∞ taking into account the definitions of Section 2.2.2 for $z(t)$, Φ and C (eqn.2.25, 2.26 and 2.28):

$$\begin{aligned} S_n^\infty &= \sum_{j=n}^{\infty} \left| C\Phi^j z(t) \right| \\ &= \sum_{j=n}^{\infty} \left\| C\Phi^j z(t) \right\|_1 \\ &= \sum_{j=n}^{\infty} \left\| C\Phi^{j-n}\Phi^n z(t) \right\|_1 \\ &= \sum_{j=0}^{\infty} \left\| C\Phi^j z(t+n) \right\|_1 \\ &\leq \left(\sum_{j=0}^{\infty} \left\| C\Phi^j \right\|_1 \right) \left\| z(t+n) \right\|_1, \end{aligned}$$

now α can be defined as

$$\alpha = \begin{cases} \sum_{j=0}^{\infty} \left\| C\Phi^j \right\|_1, & \text{if } n_{\bar{a}} > 0, \\ 0, & \text{if } n_{\bar{a}} = 0, \end{cases} \tag{2.38}$$

which can be computed with any accuracy since Φ^j decays exponentially to zero. This definition can be used to write

$$S_n^\infty \leq \alpha \left\| z(t+n) \right\|_1.$$

Furthermore, as a consequence of the definition of $z(t)$ in eqn.2.25,

$$\left\| z(t+n) \right\|_1 = \sum_{j=n-n_{\bar{a}}+1}^{n} \left| e(t+N+j|t) \right|,$$

which leads to

$$\sum_{j=0}^{\infty} \left| e(t+N+j|t) \right| \leq \sum_{j=0}^{n-1} \left| e(t+N+j|t) \right| + \alpha \sum_{j=n-n_{\bar{a}}+1}^{n} \left| e(t+N+j|t) \right|, \tag{2.39}$$

for all $n \geq 0$, as far as the constraints $\widetilde{\boldsymbol{y}} = \widetilde{\boldsymbol{w}}$ (eqn.2.22) are taken into account. Now let $J_1^-(t)$ and $J_1^+(t)$ be defined as

$$
\begin{aligned}
J_1^-(t) &= \sum_{j=1}^{N-1} |e(t+j|t)| + \sum_{j=0}^{n-1} |e(t+N+j|t)| + \sum_{j=1}^{N_u} \rho(j)\, |\Delta u(t+j-1|t)|\,, \\
J_1^+(t) &= \sum_{j=1}^{N-1} |e(t+j|t)| + \sum_{j=0}^{n-1} |e(t+N+j|t)| + \alpha \sum_{j=n-n_{\tilde{a}}+1}^{n} |e(t+N+j|t)| \qquad (2.40) \\
&+ \sum_{j=1}^{N_u} \rho(j)\, |\Delta u(t+j-1|t)|\,,
\end{aligned}
$$

for all $n \geq 0$. These definitions make it possible to state three different optimisation problems

① Lower bound problem:

$$\overline{\boldsymbol{\Delta u}}^{-\text{opt}} = \arg\min_{\boldsymbol{\Delta u}} J_1^-(t) \text{ subject to } \left\{\boldsymbol{P\Delta u} \leq \boldsymbol{r}, \widetilde{\boldsymbol{G}}\boldsymbol{\Delta u} = \widetilde{\boldsymbol{w}} - \widetilde{\boldsymbol{f}}\right\}.$$

② Infinite horizon problem:

$$\overline{\boldsymbol{\Delta u}}^{\text{opt}} = \arg\min_{\boldsymbol{\Delta u}} J_1(t) \text{ subject to } \left\{\boldsymbol{P\Delta u} \leq \boldsymbol{r}, \widetilde{\boldsymbol{G}}\boldsymbol{\Delta u} = \widetilde{\boldsymbol{w}} - \widetilde{\boldsymbol{f}}\right\}.$$

③ Upper bound problem:

$$\overline{\boldsymbol{\Delta u}}^{+\text{opt}} = \arg\min_{\boldsymbol{\Delta u}} J_1^+(t) \text{ subject to } \left\{\boldsymbol{P\Delta u} \leq \boldsymbol{r}, \widetilde{\boldsymbol{G}}\boldsymbol{\Delta u} = \widetilde{\boldsymbol{w}} - \widetilde{\boldsymbol{f}}\right\}.$$

Remark 2.20 Analogous unconstrained problems can be posed removing the general inequality constraints $\boldsymbol{P\Delta u} \leq \boldsymbol{r}$. ❑❑❑

Remark 2.21 As already remarked, the equality constraints $\widetilde{\boldsymbol{G}}\boldsymbol{\Delta u} = \widetilde{\boldsymbol{w}} - \widetilde{\boldsymbol{f}}$ are redundant in Problem ②. ❑❑❑

Remark 2.22 Since these three problems take into account *the same* constraints, a given point $\boldsymbol{\Delta u}$ in the decision variables space is either feasible or infeasible for all of them. Hence the value of these cost functions can be compared for any $\boldsymbol{\Delta u}$. ❑❑❑

Remark 2.23 Problems ① and ③ consider finite-horizon cost functions, whereas in Problem ② an infinite horizon one is to be minimised. Thus, Problems ① and ③ can be exactly solved for every n, since they can be written as finite-dimensional LP problems.

❑❑❑

Remark 2.24 Notice that $J_1^-(t)$ and $J_1^+(t)$ converge to $J_1(t)$ as $n \to \infty$:

$$\lim_{n\to\infty} J_1^-(t) = J_1(t),$$
$$\lim_{n\to\infty} J_1^+(t) = J_1(t),$$

and obviously, as a consequence of this,

$$\lim_{n\to\infty} \overline{\boldsymbol{\Delta u}^{-\text{opt}}(t)} = \overline{\boldsymbol{\Delta u}^{\text{opt}}(t)}, \qquad \lim_{n\to\infty} \overline{J^{-\text{opt}}(t)} = \overline{J^{\text{opt}}(t)},$$
$$\lim_{n\to\infty} \overline{\boldsymbol{\Delta u}^{+\text{opt}}(t)} = \overline{\boldsymbol{\Delta u}^{\text{opt}}(t)}, \qquad \lim_{n\to\infty} \overline{J^{+\text{opt}}(t)} = \overline{J^{\text{opt}}(t)}.$$

❑❑❑

It is found convenient here to introduce here an alternative notation for the cost function evaluation:

$$J\left(\boldsymbol{\Delta u}\right) = J(t)|_{\boldsymbol{\Delta u}(t)=\boldsymbol{\Delta u}} \,.$$

Remark 2.25 For a feasible $\boldsymbol{\Delta u}$ (Remark 2.22) it follows that

$$J_1^-\left(\boldsymbol{\Delta u}\right) \leq J_1\left(\boldsymbol{\Delta u}\right) \leq J_1^+\left(\boldsymbol{\Delta u}\right), \tag{2.41}$$

that is, $J_1^-(t)$ and $J_1^+(t)$ are, respectively, upper and lower bounds of $J_1(t)$. The leftmost inequality is straightforward, since any finite horizon cost function value must be lower than or equal to any infinite horizon one, as far as the weighting sequences are the same. In other words, there are infinitely many more non-negative terms in $J_1(t)$ than in $J_1^-(t)$. The rightmost inequality follows directly from eqn.2.39. ❑❑❑

Taking into account these remarks, it is now possible to formulate and prove some theorems which are useful to find an implementable solution of the *a priori* infinite dimensional LP problem related to the infinite prediction horizon used in Problem ②.

Theorem 2.3 *The optimal values of the cost functions $J_1^-(t)$, $J_1(t)$ and $J_1^+(t)$ satisfy the following inequalities:*

$$J_1^-\left(\overline{\boldsymbol{\Delta u}^{-\text{opt}}}\right) \leq J_1\left(\overline{\boldsymbol{\Delta u}^{\text{opt}}}\right) \leq J_1^+\left(\overline{\boldsymbol{\Delta u}^{+\text{opt}}}\right). \tag{2.42}$$

Remark 2.26 Notice that as $n \to \infty$ the leftmost and the rightmost values of eqn.2.42 converge to $J_1\left(\overline{\boldsymbol{\Delta u}^{\text{opt}}}\right)$, as stated in Remark 2.24. □□□

Proof: In the light of eqn.2.41 it comes out that

$$J_1^-\left(\overline{\boldsymbol{\Delta u}^{-\text{opt}}}\right) \overset{(1)}{\leq} J_1^-\left(\overline{\boldsymbol{\Delta u}^{\text{opt}}}\right) \overset{(2)}{\leq} J_1\left(\overline{\boldsymbol{\Delta u}^{\text{opt}}}\right) \overset{(3)}{\leq} J_1\left(\overline{\boldsymbol{\Delta u}^{+\text{opt}}}\right) \overset{(4)}{\leq} J_1^+\left(\overline{\boldsymbol{\Delta u}^{+\text{opt}}}\right), \tag{2.43}$$

where (2) and (4) follow from eqn.2.41 and the (1) and (3) are a consequence of the optimality principle according to which the optimal value is always lower than or equal to any other feasible value. ∇∇∇

Corollary 2.1 *If* $\left|J_1^+\left(\overline{\boldsymbol{\Delta u}^{+\text{opt}}}\right) - J_1^-\left(\overline{\boldsymbol{\Delta u}^{-\text{opt}}}\right)\right| < \varepsilon$, *then*

$$\left|J_1\left(\overline{\boldsymbol{\Delta u}^{+\text{opt}}}\right) - J_1\left(\overline{\boldsymbol{\Delta u}^{\text{opt}}}\right)\right| < \varepsilon.$$

(The absolute value signs can be dropped).

Proof: The proof is straightforward from eqn.2.43. ∇∇∇

Theorem 2.4 *For all $\varepsilon > 0$ (arbitrarily small) there exists an integer $n_0 \geq 0$ such that $\left|\overline{J_1^{+\text{opt}}}(t) - \overline{J_1^{-\text{opt}}}(t)\right| < \varepsilon$, for $n \geq n_0$, or*

$$\forall \varepsilon > 0,\ \exists n_0 \geq 0 : \left|J_1^+\left(\overline{\boldsymbol{\Delta u}^{+\text{opt}}}\right) - J_1^-\left(\overline{\boldsymbol{\Delta u}^{-\text{opt}}}\right)\right| < \varepsilon,\ \forall n \geq n_0. \tag{2.44}$$

Proof: As a consequence of Remark 2.24:

$$\forall \varepsilon_1 > 0,\ \exists n_1 \geq 0 : \left|J_1^-\left(\overline{\boldsymbol{\Delta u}^{+\text{opt}}}\right) - J_1\left(\overline{\boldsymbol{\Delta u}^{\text{opt}}}\right)\right| < \varepsilon_1,\ \forall n \geq n_1,$$
$$\forall \varepsilon_2 > 0,\ \exists n_2 \geq 0 : \left|J_1^+\left(\overline{\boldsymbol{\Delta u}^{+\text{opt}}}\right) - J_1\left(\overline{\boldsymbol{\Delta u}^{\text{opt}}}\right)\right| < \varepsilon_2,\ \forall n \geq n_2,$$

now consider $\varepsilon_1 = \varepsilon/2$, $\varepsilon_2 = \varepsilon/2$ and $n_0 = \max\{n_1, n_2\}$ thus, for all $n \geq n_0$:

$$\begin{aligned}\left|\overline{J_1^{+\text{opt}}} - \overline{J_1^{-\text{opt}}}\right| &= \left|J_1^+\left(\overline{\Delta u^{+\text{opt}}}\right) - J_1\left(\overline{\Delta u^{\text{opt}}}\right) + J_1\left(\overline{\Delta u^{\text{opt}}}\right) - J_1^-\left(\overline{\Delta u^{-\text{opt}}}\right)\right| \\ &\leq \left|J_1^+\left(\overline{\Delta u^{+\text{opt}}}\right) - J_1\left(\overline{\Delta u^{\text{opt}}}\right)\right| + \left|J_1^-\left(\overline{\Delta u^{-\text{opt}}}\right) - J_1\left(\overline{\Delta u^{\text{opt}}}\right)\right| \\ &< \varepsilon_1 + \varepsilon_2 \\ &= \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\ &= \varepsilon,\end{aligned}$$

what completes the proof. ∇∇∇

Corollary 2.2 *In the unconstrained case, for all $\varepsilon > 0$ there exists an integer $n_0 \geq 0$ such that $\left|J_1^{+\text{opt}}(t) - J_1^{-\text{opt}}(t)\right| < \varepsilon$, for $n \geq n_0$, or*

$$\forall \varepsilon > 0,\ \exists n_0 \geq 0 : \left|J_1^+\left(\Delta u^{+\text{opt}}\right) - J_1^-\left(\Delta u^{-\text{opt}}\right)\right| < \varepsilon,\ \forall n \geq n_0.$$

Proof: This corollary directly follows from Theorem 2.4, since the unconstrained optimisation problems are a particular case of the constrained optimisation counterparts, in which the inequality constraints $\boldsymbol{P}\Delta\boldsymbol{u} < \boldsymbol{r}$ are empty. ∇∇∇

Remark 2.27 For "small enough" ε: $J_1^+\left(\overline{\Delta u^{+\text{opt}}}\right) \approx J_1^-\left(\overline{\Delta u^{-\text{opt}}}\right)$, then (Corollary 2.1) $J_1\left(\overline{\Delta u^{+\text{opt}}}\right) \approx J_1\left(\overline{\Delta u^{\text{opt}}}\right)$, and finally $\overline{\Delta u^{\text{opt}}} \approx \overline{\Delta u^{+\text{opt}}}$ due to the convexity of $J_1(t)$. □□□

The last remark can be used to derive an algorithm to find $\overline{\Delta u^{\text{opt}}}$ to any given accuracy. The algorithm can be set up as follows:

1. Choose a small relative error ε_r
2. Choose[7] $[n+1]$
3. Solve the optimisation Problems ① and ③ to obtain $\overline{J_1^{-\text{opt}}}$, $\overline{\Delta u^{-\text{opt}}}$, $\overline{J_1^{+\text{opt}}}$ and $\overline{\Delta u^{+\text{opt}}}$

[7]The reason for working with $[n+1]$ rather than n is clarified in Section 2.3.2.1.

4. **if** $\overline{J_1^{+\text{opt}}} - \overline{J_1^{-\text{opt}}} < \varepsilon_r \cdot \overline{J_1^{-\text{opt}}}$

   then

   $\overline{J_1^{\text{opt}}} := \overline{J_1^{+\text{opt}}}$;

   $\overline{\Delta u^{\text{opt}}} := \overline{\Delta u^{+\text{opt}}}$;

   return

   else

   Increase $[n+1]$;

   goto 3;

   endif

It is worth pointing out a few comments about the algorithm depicted above:

- The stopping criterion used in step 4 can be modified so as to consider the absolute error on the solution:

$$\overline{J_1^{+\text{opt}}} - \overline{J_1^{-\text{opt}}} < \varepsilon_a$$

  for a given $\varepsilon_a > 0$, or the absolute error on the first postulated control move:

$$\left|\overline{\Delta u^{+\text{opt}}}(t|t) - \overline{\Delta u^{-\text{opt}}}(t|t)\right| < \varepsilon_{\Delta u}$$

  for a given $\varepsilon_{\Delta u} > 0$, since only the first computed control move is used in the receding-horizon strategy. These two considerations can be very useful to reduce the number of iterations. However, it must remarked that these two conditions, especially the latter, do not guarantee that $\overline{J_1^{+\text{opt}}}(t) \approx \overline{J_1^{-\text{opt}}}(t)$, and hence the optimal cost function sequence might not satisfy the monotonicity property, which is essential for the stability proof.

- Theorem 2.4 does not provide any bound or estimate on how big n_0 can become. An exponential update $[n+1]$:=$2[n+1]$ is advised to decrease the number of iterations. However, notice that the number of variables and constraints in the LP problems to be solved depends on n.

- At each iteration two (finite-dimensional) LP problems must be solved hence, if many iterations are needed, the computation time of a single control move can be too high for practical implementation, especially if a very large n is required for convergence.

- It must be taken into account that the optimisation is to be performed on line for a number samples, and the information about what $[n+1]$ has been required for convergence at the previous sample is available. At the sample $i+1$, $[n+1]$ can be initialised as *half* the value required for convergence at the previous one i (with a minimum, for example 10):

$$[n+1]^{(i+1)} := \max\left\{\frac{[n+1]^{(i)}}{2}, 10\right\}. \tag{2.45}$$

  In this fashion, it is possible to reduce the number of iterations at each sample, since unnecessarily small values of n are not used after the first one. Moreover, the dimension of the LP problems can decrease from sample to sample. This procedure can significantly reduce the computation time required by the controller.

The following section is concerned with the solution of Problems ① and ③ using LP tools.

2.3.2.1 Solution of the upper bound and the lower bound problems

Both the lower bound and upper bound problems can be solved using standard LP methods, quite similarly as done for the case of CRHPC_1 in Section 2.3.1. The objective here is to find the minima of $J_1^-(t)$ and $J_1^+(t)$ as defined in eqn.2.40 subject to the constraints reported in the definition of Problems ① and ③. In order to solve these problems, $\sigma(j) \geq 0$, $\varphi(j) \geq 0$ and $\beta(j) \geq 0$ are defined such that

$$\begin{array}{rcccl} -\sigma(j) & \leq & e(t+j|t) & \leq & \sigma(j), \quad 1 \leq j \leq N, \\ -\varphi(j) & \leq & e(t+N+j|t) & \leq & \varphi(j), \quad 0 \leq j \leq n, \\ -\beta(j) & \leq & \Delta u(t+j-1|t) & \leq & \beta(j), \quad 1 \leq j \leq N_u, \\ 0 & \leq & \sum_{j=1}^{N} \mu(j)\sigma(j) + \sum_{j=0}^{n} \kappa(j)\varphi(j) + \sum_{j=1}^{N_u} \rho(j)\beta(j) & \leq & \Psi, \end{array}$$

where[8]

$$\mu(j) = \begin{cases} 1 & \text{if } 0 < j < N, \\ 0 & \text{if } j = N, \end{cases}$$

and $\kappa(j)$ takes different values for the lower bound and the upper bound problems (① and ③ respectively):

$$① \; \kappa(j) = \begin{cases} 1 & \text{if } 0 \leq j < n, \\ 0 & \text{if } j = n, \end{cases}$$

$$③ \; \kappa(j) = \begin{cases} 1 & \text{if } 0 \leq j \leq n - n_{\bar{a}}, \\ 1+\alpha & \text{if } n - n_{\bar{a}} < j \leq n-1, \\ \alpha & \text{if } j = n, \end{cases}$$

with α as defined in eqn.2.38. In the upper bound problem, if $n_{\bar{a}} = 0$ (antistable system), it makes no difference to define $\kappa(n)$ as α (which is zero in such a case) or 1, since the predicted error $e(t+N+n|t)$ must be zero due to the end-point equality constraints.

Remark 2.28 In the upper bound problem, if $n < n_{\bar{a}}$, the coefficients $\kappa(j)$ and $\mu(j)$ overlap, and a few small modifications are required (see the QGPC$_1^\infty$ formulation in Section 2.3.3 for an example of this). Hence, it is assumed that $n \geq n_{\bar{a}}$ unless otherwise explicitly specified. □□□

The minimisation of J_1^- and J_1^+ is equivalent to that of Ψ taking the appropriate value of $\kappa(j)$ in each case.

[8]Notice that the first term of J_1^- and J_1^+ sums the absolute value of the errors up to $N-1$ and not N, hence $\mu(N) = 0$. The definition $\mu(N) = 0$ guarantees that $|e(t+N|t)|$ is counted only once in the cost function.

In order to write the LP problems in a more convenient form, let e_n be defined as

$$e_n(t) = \left[\; e(t+N|t) \quad e(t+N+1|t) \quad \dots \quad e(t+N+n|t) \;\right]^{\mathrm{T}},$$

that is, $e_n(t)$ is the vector of predicted errors $e(t+N+j|t)$ for $j = 0, 1, \dots, n$. Problems ① and ③ can now be written as

$$\min_{\Psi,\sigma,\varphi,\beta,\Delta u} \Psi \text{ subject to } \begin{cases} \sigma \geq G\Delta u + f - w, \\ \sigma \geq -G\Delta u - f + w, \\ \varphi \geq -e_n, \\ \varphi \geq e_n, \\ \beta \geq \Delta u, \\ \beta \geq -\Delta u, \\ \Psi \geq \mu^{\mathrm{T}}\sigma + \kappa^{\mathrm{T}}\varphi + \rho^{\mathrm{T}}\beta, \\ P\Delta u \leq r, \\ \widetilde{G}\Delta u = \widetilde{w} - \widetilde{f}, \\ \sigma \geq 0, \varphi \geq 0,\ \beta \geq 0,\ \Psi \geq 0, \end{cases}$$

where

$$\varphi = \left[\; \varphi(0) \quad \varphi(1) \quad \dots \quad \varphi(n) \;\right]^{\mathrm{T}},$$

and μ, σ, ρ and β are defined in eqn.2.35. The iterative algorithm presented above works with $[n+1]$ instead of n because the dimension of φ is $(n+1) \times 1$ and not $n \times 1$. Finally, κ is a weighting vector which takes two different values for Problems ① and ③:

$$\text{①}\ \kappa = \left[\begin{array}{cccc} 1 & \overset{(n)}{\dots} & 1 & 0 \end{array}\right]^{\mathrm{T}},$$
$$\text{③}\ \kappa = \left[\begin{array}{cccc} 1 & \overset{(n)}{\dots} & 1 & 0 \end{array}\right]^{\mathrm{T}} + \left[\begin{array}{cccccc} 0 & \overset{(n-n_{\bar{a}}+1)}{\dots} & 0 & \alpha & \overset{(n_{\bar{a}})}{\dots} & \alpha \end{array}\right]^{\mathrm{T}}.$$

Taking into account that the end-point constraints $\widetilde{y} = \widetilde{w}$ imply eqn.2.27, it is possible to write the vector $e_n(t)$ in terms of $z(t)$:

$$e_n(t) = \begin{bmatrix} Cz(t) \\ C\Phi z(t) \\ \vdots \\ C\Phi^n z(t) \end{bmatrix} = \Gamma z(t),$$

with

$$\Gamma = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^n \end{bmatrix},$$

where $z(t)$, Φ and C are defined in eqn.2.25, 2.28 and 2.26 respectively. In addition, according to the definition of $e(t)$ in eqn.2.17, $z(t) = He(t)$ for

$$H = \left[\begin{array}{cc} 0_{N,N-n_{\tilde{a}}} & I_{n_{\tilde{a}}} \end{array}\right].$$

Hence, the constraints $\varphi \geq -e_n$ and $\varphi \geq e_n$ can be written in a more suitable form, since

$$e_n = \Gamma z = \Gamma He = \Gamma H (w - f - G\Delta u),$$

thus

$$\varphi \geq -e_n \Rightarrow \varphi \geq -\Gamma H (w - f) + \Gamma HG\Delta u,$$
$$\varphi \geq e_n \quad \Rightarrow \varphi \geq \Gamma H (w - f) - \Gamma HG\Delta u.$$

Finally, these LP problems are converted to the standard form

$$\min_{x} c^{\mathrm{T}} x \text{ subject to } Ax \leq b,\ \sigma \geq 0,\ \varphi \geq 0,\ \beta \geq 0,\ \Psi \geq 0,$$

with

$$x = \left[\begin{array}{c} \Delta u \\ \hline \sigma \\ \hline \varphi \\ \hline \beta \\ \hline \Psi \end{array}\right], \quad c = \left[\begin{array}{c} 0 \\ \hline 0 \\ \hline 0 \\ \hline 0 \\ \hline 1 \end{array}\right],$$

and

$$A = \left[\begin{array}{c|c|c|c|c} G & -I & 0 & 0 & 0 \\ -G & -I & 0 & 0 & 0 \\ \hline \Gamma HG & 0 & -I & 0 & 0 \\ -\Gamma HG & 0 & -I & 0 & 0 \\ \hline I & 0 & 0 & -I & 0 \\ -I & 0 & 0 & -I & 0 \\ \hline 0^{\mathrm{T}} & \mu^{\mathrm{T}} & \kappa^{\mathrm{T}} & \rho^{\mathrm{T}} & -1 \\ \hline P & 0 & 0 & 0 & 0 \\ \hline \widetilde{G} & 0 & 0 & 0 & 0 \\ -\widetilde{G} & 0 & 0 & 0 & 0 \end{array}\right], \quad b = \left[\begin{array}{c} -f + w \\ f - w \\ \hline \Gamma H (-f + w) \\ \Gamma H (f - w) \\ \hline 0 \\ 0 \\ \hline 0 \\ \hline r \\ \hline -\widetilde{f} + \widetilde{w} \\ \widetilde{f} - \widetilde{w} \end{array}\right]$$

2.3.3 Quasi-infinite horizon 1-norm GPC

As remarked above, the implementation of the GPC_1^∞ can involve a high computational burden due to the necessity of solving two LP problems at each iteration, and there is no estimate on how many iterations will be needed. In addition, the dimension of these problems increases with n. The *Quasi-Infinite Horizon 1-norm GPC* (QGPC_1^∞) presented in this section is based on minimising an upper bound of the GPC_1^∞ cost function, obtained from eqn.2.39 with $n = 0$:

$$\sum_{j=0}^{\infty} |e(t+N+j|t)| \leq \alpha \sum_{j=N-n_{\tilde{a}}+1}^{N} |e(t+j|t)|, \tag{2.46}$$

where α is defined in eqn.2.38. Thus, this is the simplest possible upper bound of the infinite horizon cost function regarding the dimension of the LP problem to be solved.

The main point on this approach is the fact that only one LP problem must be solved, and hence the iterative procedure depicted above can be avoided. However, as shown in the sequel, this method does not guarantee the stability of the nominal closed-loop system, though it is "very unlikely" that instability results.

The motivation for using an upper bound of the infinite horizon problem comes from the intuitive idea that the stability problems of the GPC are somehow related to the use of a finite prediction horizon. In fact, any finite horizon cost function is a lower bound of the infinite horizon one (unless particular system-dependent weighting sequences are chosen), and that can be one of the reasons for the poor GPC stabilising properties. A better behaviour might arise by using a cost function which bounds the infinite horizon problem from above. Intuitively, if an upper bound is minimised, the true infinite horizon cost function would be below the upper bound optimal value and, thus, the infinite horizon cost function is, if not minimised, at least bounded from above at its optimal point.

Taking into account all these considerations, the QGPC_1^∞ proposed here computes the optimal control move vector as the solution of the optimisation problem

$$\overline{\Delta \boldsymbol{u}^{\mathrm{opt}}} = \arg\min_{\Delta \boldsymbol{u}} J_1(t) \text{ subject to } \left\{ \boldsymbol{P}\Delta\boldsymbol{u} \leq \boldsymbol{r}, \widetilde{\boldsymbol{G}}\Delta\boldsymbol{u} = \widetilde{\boldsymbol{w}} - \widetilde{\boldsymbol{f}} \right\}, \tag{2.47}$$

with

$$J_1(t) = \sum_{j=1}^{N-1} |e(t+j|t)| + \alpha \sum_{j=N-n_{\bar{a}}+1}^{N} |e(t+j|t)| + \sum_{j=1}^{N_u} \rho(j)\, |\Delta u(t+j-1|t)|, \tag{2.48}$$

which is the same as $J_1^+(t)$ of eqn.2.40 with $n = 0$.

The LP formulation used in Section 2.3.2.1 to minimise $J_1^+(t)$ must be modified in order to solve the QGPC$_1^\infty$ problem, since the vectors $\boldsymbol{\mu}$ and $\boldsymbol{\kappa}$ overlap in the latter case. If the variables $\sigma(j) \geq 0$ and $\beta(j) \geq 0$ are introduced, the problem of minimising Ψ subject to:

$$\begin{array}{rcccl} -\sigma(j) & \leq & e(t+j|t) & \leq & \sigma(j), \quad 1 \leq j \leq N, \\ -\beta(j) & \leq & \Delta u(t+j-1|t) & \leq & \beta(j), \quad 1 \leq j \leq N_u, \\ 0 & \leq & \sum_{j=1}^{N} \mu(j)\sigma(j) + \sum_{j=1}^{N_u} \rho(j)\beta(j) & \leq & \Psi, \end{array}$$

with

$$\mu(j) = \begin{cases} 1 & \text{if } 0 \leq j \leq N - n_{\bar{a}}, \\ 1+\alpha & \text{if } N - n_{\bar{a}} < j \leq N-1, \\ \alpha & \text{if } j = N, \end{cases}$$

is equivalent to the problem of eqn.2.47. If $n_{\bar{a}} = 0$ (antistable system) $\mu(N)$ can be defined to be either α (which is zero in that case) or 1, since the predicted error $e(t+N|t)$ is zero due to the end-point equality constraints.

As done in Section 2.3.2.1, this problem can be converted to the standard form as

$$\min_{\boldsymbol{x}} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \text{ subject to } \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b},\ \boldsymbol{\sigma} \geq \boldsymbol{0},\ \boldsymbol{\beta} \geq \boldsymbol{0},\ \Psi \geq 0,$$

with

$$\boldsymbol{x} = \left[\begin{array}{c} \Delta\boldsymbol{u} \\ \hline \boldsymbol{\sigma} \\ \hline \boldsymbol{\beta} \\ \hline \Psi \end{array} \right], \qquad \boldsymbol{c} = \left[\begin{array}{c} \boldsymbol{0} \\ \hline \boldsymbol{0} \\ \hline \boldsymbol{0} \\ \hline 1 \end{array} \right],$$

and

$$A = \left[\begin{array}{c|c|c|c} G & -I & 0 & 0 \\ -G & -I & 0 & 0 \\ \hline I & 0 & -I & 0 \\ -I & 0 & -I & 0 \\ \hline 0^{\mathrm{T}} & \mu^{\mathrm{T}} & \rho^{\mathrm{T}} & -1 \\ \hline P & 0 & 0 & 0 \\ \hline \tilde{G} & 0 & 0 & 0 \\ -\tilde{G} & 0 & 0 & 0 \end{array}\right], \qquad b = \left[\begin{array}{c} -f+w \\ f-w \\ \hline 0 \\ 0 \\ \hline 0 \\ \hline r \\ \hline -\tilde{f}+\tilde{w} \\ \tilde{f}-\tilde{w} \end{array}\right],$$

where μ, σ, β and ρ are defined in eqn.2.35.

2.3.4 1-norm stability theorems

In this section stability theorems for the CRHPC$_1$, the GPC$_1^\infty$ and the QGPC$_1^\infty$ are provided. Some of them are 1-norm versions of those of Section 2.2.3.

Theorem 2.5 *For any stabilisable and detectable system, the CRHPC$_1$ leads to a stable closed-loop system if*

(i) $\mu(N) \geq \mu(N-1) \geq \cdots \geq \mu(1)$,

(ii) $\rho(N_u) \geq \rho(N_u - 1) \geq \cdots \geq \rho(1) \geq 0$, *and*

(iii) $m = \max\{n_a, n_b - 1\} + 1$.

Proof: The proof is completely analogous to that of Theorem 2.1, replacing the "squares" by absolute value signs. ∇∇∇

All the remarks made about the CRHPC are valid for the 1-norm counterpart.

Theorem 2.6 *For any stabilisable and detectable system, the* GPC$_1^\infty$ *is stabilising if*

(i) $\rho(N_u) \geq \rho(N_u - 1) \geq \cdots \geq \rho(1) \geq 0$, *and*

(ii) $N_u > n_{\tilde{a}}$.

Proof: The proof provided for Theorem 2.2 can be written replacing the "squares" by absolute value signs. ∇∇∇

Again, all the remarks made about the GPC^∞ are valid for the 1-norm counterpart.

Theorem 2.7 *For any stabilisable and detectable system, the* QGPC_1^∞ *is stabilising if*

(i) $\rho(N_u) \geq \rho(N_u - 1) \geq \cdots \geq \rho(1) \geq 0$, *and*

(ii) $N_u = n_{\tilde{a}} + 1$.

Proof: In that case, the QGPC_1^∞ is equivalent to both the GPC^∞ and the GPC_1^∞ since, in these three controllers, for $N_u = n_{\tilde{a}} + 1$, all the available degrees of freedom are used to enforce the $n_{\tilde{a}} + 1$ end-point equality constraints $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{w}}$. Hence the vectors $\overline{\boldsymbol{\Delta u}^{\text{opt}}}(t)$ computed by these controllers are the same and stability is guaranteed by Theorems 2.2 and 2.6. ∇∇∇

Theorem 2.8 *For any stabilisable and detectable antistable*[9] *system, the* QGPC_1^∞ *is stabilising if*

(i) $\rho(N_u) \geq \rho(N_u - 1) \geq \cdots \geq \rho(1) \geq 0$, *and*

(ii) $N_u > n_{\tilde{a}} = n_a$.

Proof: For antistable systems $\bar{A}(q^{-1}) = 1$, $\tilde{\boldsymbol{y}} = \boldsymbol{y}$ and $\tilde{\boldsymbol{w}} = \boldsymbol{w}$ and then the $n_{\tilde{a}} + 1$ end-point equality constraints on the unstable part of the output are, in fact, $n_a + 1$ equality constraints on the whole output. Due to these end-point constraints, the predicted errors $e(t + N + j|t)$ are zero for all $j \geq 0$, what makes the cost functions of the GPC_1^∞ (eqn.2.36) and the QGPC_1^∞ (eqn.2.48) identical. Hence the QGPC_1^∞

[9] A system is said to be antistable if all of its poles are unstable, or $n_{\bar{a}} = 0$, $n_a = n_{\tilde{a}}$.

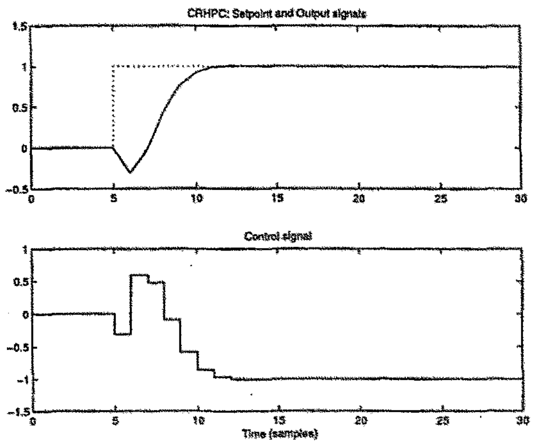
is equivalent to the GPC_1^∞ and stability of the closed-loop system is guaranteed by Theorem 2.6. ∇∇∇

Remark 2.29 In addition, in this situation, the $QGPC_1^\infty$ and the GPC_1^∞ closely resemble the $CRHPC_1$, because the end-point constraints on the unstable part of the output turn out to be equality constraints on the whole output. Notice, however, that the $CRHPC_1$ may need more constraints than the $GPC_1^\infty/QGPC_1^\infty$, since m (Theorem 2.5) must be greater than the numerator order: $m = \max\{n_a, n_b - 1\} + 1$. This condition is not necessary in the $GPC^\infty/QGPC_1^\infty$ formulations, since the prediction horizon N is defined in eqn.2.14 such that the dynamics caused by the numerator reach the steady state by time $t + N$. □□□

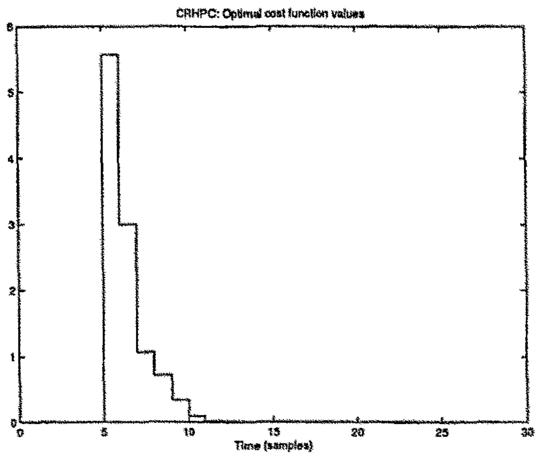
2.4 Illustrative examples

In this section the properties of the stabilising controllers CRHPC, $CRHPC_1$, GPC^∞, GPC_1^∞ and $QGPC_1^\infty$ are analysed by means of simulation using the benchmark systems introduced in Appendix A. The lower costing horizons for the GPC^∞, the GPC_1^∞ and the $QGPC_1^\infty$ have been chosen as $N_1 = 1$, since this tuning knob does not affect the stability properties of these controllers. In addition, it is worth pointing out that this chapter is only concerned with the nominal properties. Thus the true system and the internal model for predictions are identical for all the experiments of this section. Hence, as the polynomial T does not affect the results (see Chapter 3), $T(q^{-1}) = 1$ is assumed.

In the implementation of the GPC_1^∞ used throughout this section, the iterative algorithm presented in Section 2.3.2 has been used with a relative error $\varepsilon_r = 10^{-2}$, *i.e.* the difference between the lower and the upper bounds of the infinite horizon cost function must be less than the 1% of the lower bound optimum value. The n parameter has been initialised to $[n + 1] := 10$ (at the first sample), whereas the sentence of eqn.2.45 has been used from the second sampling instant on.



(a) Input/Ouput responses for the (2-norm) CRHPC and GPC$^{\infty}$



(b) Optimal cost function values for the (2-norm) CRHPC and GPC$^{\infty}$

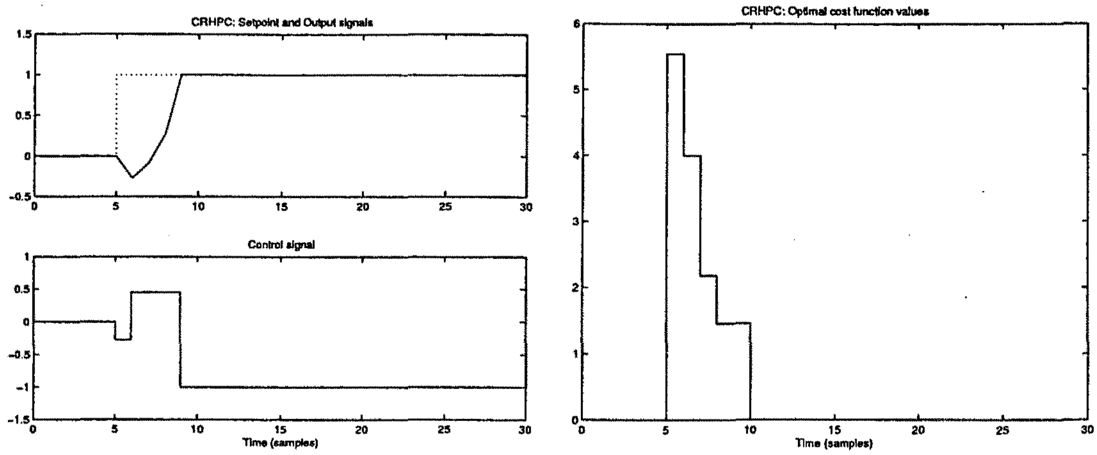
Figure 2.1: Closed-loop behaviour of the (2-norm) CRHPC: $[N, m, \mu, \rho] = [4, 3, 1, 1]$ and GPC$^{\infty}$: $[N_u, \rho] = [5, 1]$

2.4.1 An unstable GPC example

First of all, the controllers have been tested on the system of eqn.A.1 (Section A.1) which is reported in (Bitmead *et al.*, 1990) to cause stability problems with the standard finite horizon GPC:

$$G(q^{-1}) = \frac{q^{-1}B(q^{-1})}{A(q^{-1})} = \frac{q^{-1} - 1.999q^{-2}}{1 - 4q^{-1} + 4q^{-2}}.$$

The CRHPC and CRHPC$_1$ have been tried using the tuning parameters $N = 4$, $m = 3$, $\mu(j) = 1$, for all j and $\rho(j) = 1$ for all j. With such a choice, m, $\mu(j)$ and $\rho(j)$ satisfy the conditions of Theorems 2.1 and 2.5. Thus, both controllers should provide with a stable closed-loop system. Notice that $N_u = N + 1 = 5$ is greater than the number of equality constraints ($m = 3$), which is a necessary condition for the solvability of the optimisation problem.



(a) Input/Ouput responses for the CRHPC_1, GPC_1^∞ and QGPC_1^∞

(b) Optimal cost function values for the CRHPC_1, GPC_1^∞ and QGPC_1^∞

Figure 2.2: Closed-loop behaviour of the CRHPC_1: $[N, m, \mu, \rho] = [4, 3, 1, 1]$ and $\mathrm{GPC}_1^\infty/\mathrm{QGPC}_1^\infty$: $[N_u, \rho] = [5, 1]$

Fig.2.1 and 2.2 show the closed-loop behaviour obtained with the 2-norm and the 1-norm controllers. In the figures, a dotted line is used for the setpoint, which changes from 0 to 1 at the fifth sample. It is noticed that both controllers lead to a stable closed-loop system with monotonically non-increasing optimal cost function values. Only when the setpoint change occurs (at the fifth sample) does the cost function raise, otherwise it decreases and settles down to zero. Notice also that the difference between the 2-norm and the 1-norm versions is remarkably small, either as the input/output responses or the cost functions are concerned. It must be pointed out, however, that the comparison between the closed-loop behaviour provided by the 1-norm and 2-norm implementations is not strictly appropriate, since the weights μ and ρ apply to absolute values in the former, but to squares in the latter.

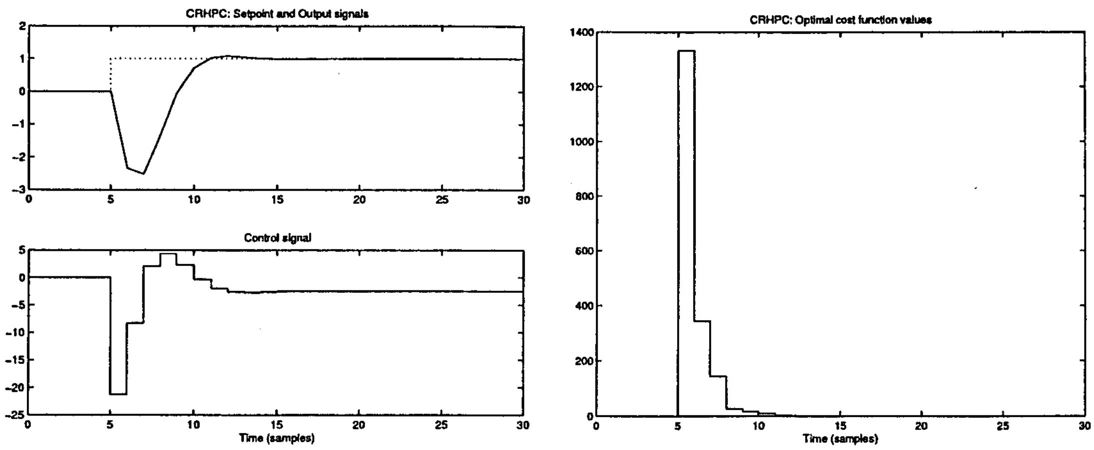
The GPC^∞ and the GPC_1^∞ have been tested on this system using the tuning parameters $N_u = 5$ and $\rho(j) = 1$ for all j. Hence, the degrees of freedom of these controllers are the same as for the CRHPC and the CRHPC_1 used above. In this case, the GPC^∞ turns out to be identical to the CRHPC, and the same goes to GPC_1^∞

and the CRHPC$_1$. This fact is not surprising, as the system used for the experiments is antistable and thus, the equality constraints on the unstable part of the output become, for both the GPC$^\infty$ and the GPC$_1^\infty$, equality constraints on the whole output. However, the fact that both controllers (the CRHPC and the GPC$^\infty$) are identical in this case is not straightforward. On the one hand, with the tuning knobs $[N, m, \mu(j), \rho(j)] = [4, 3, 1, 1]$ the CRHPC (CRHPC$_1$) is equivalent to the problem of minimising eqn.2.3 (eqn.2.34) with $[N_1, N_y, N_2, N_u, \mu(j), \rho(j), \gamma] = [1, 5, 7, 5, 1, 1, 0]$. On the other hand, the GPC$^\infty$ (GPC$_1^\infty$), taking into account that the end-point constraints affect the whole output of the model, can be posed as the minimisation of eqn.2.3 (eqn.2.34) with $[N_1, N_y, N_2, N_u, \mu(j), \rho(j), \gamma] = [1, 6, 8, 5, 1, 1, 0]$. Hence the constraint and upper costing horizons N_y and N_2 are not the same for the CRHPC and the GPC$^\infty$. Despite that, it can be easily shown that, for this particular case, the equality constraints enforced by the CRHPC and the GPC lead to $y(t+j|t) = w(t|t)$ **for all** $j \geq 4$, which is reason why these two controllers become identical.

Remark 2.30 One might be tempted to think that the CRHPC (CRHPC$_1$) and the GPC$^\infty$ (GPC$_1^\infty$) are *equivalent for all antistable systems* when the degrees of freedom (N_u) of these controllers are the same. This is only true when the number of open-loop zeros is lower than or equal to the number of open-loop poles, or $n_b - 1 \leq n_a$, since, in that case, both the CRHPC and the GPC$^\infty$ use the same number of end-point equality constraints: $m = \max\{n_a, n_b - 1\} + 1 = n_a + 1$ for the CRHPC and $n_{\tilde{a}} + 1 = n_a + 1$ for the GPC$^\infty$. ❑❑❑

Finally, as a consequence of Theorem 2.8, the QGPC$_1^\infty$ with the tuning knobs $[Nu, \rho(j)] = [5, 1]$ is equivalent to the GPC$_1^\infty$ and, consequently, to the CRHPC$_1$. Hence the closed-loop behaviour for the QGPC$_1^\infty$ is as shown in Fig.2.2.

The system used in this section can be regarded as quite a pathological case, since all its roots (poles and zeroes) are unstable and, moreover, it possesses a near pole-zero



(a) Input/Ouput responses for the (2-norm) CRHPC

(b) Optimal cost function values for the (2-norm) CRHPC

Figure 2.3: Closed-loop behaviour of the (2-norm) CRHPC: $[N, m, \mu, \rho] = [4, 3, 1, 1]$

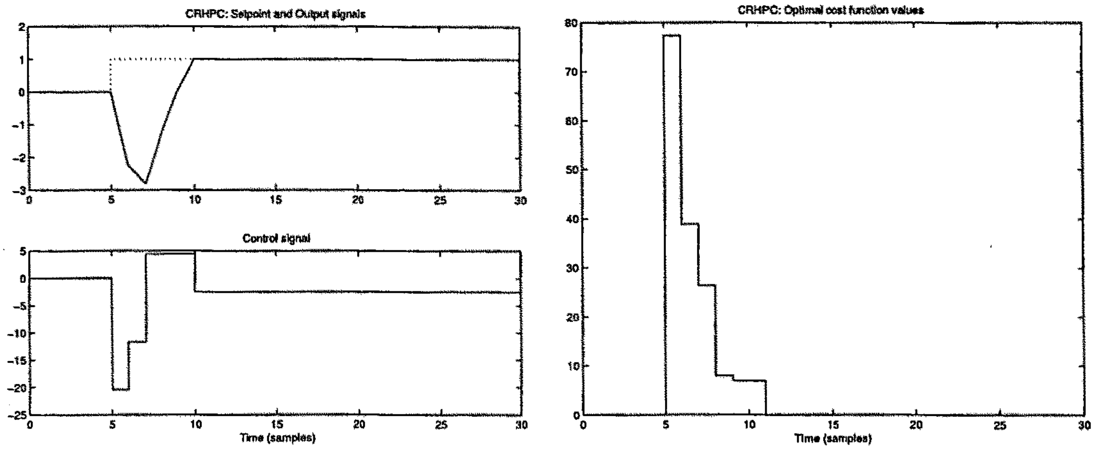
cancellation which results on the system being almost undetectable. The stabilising approaches have been proved successful even in such a situation. In fact, antistable systems are the only ones for which the QGPC$_1^\infty$ is proved to be stabilising for all possible values of the control horizon $N_u > n_{\tilde{a}}$. The next few sections are devoted to illustrate the behaviour of these controllers in front of more typical situations.

2.4.2 Non-minimum phase stable system

Here, the controllers are tested on the system provided in eqn.A.4 of Section A.2:

$$G(q^{-1}) = \frac{q^{-1}B(q^{-1})}{A(q^{-1})} = \frac{q^{-1}(0.1098 - 0.1232q^{-1})}{1 - 1.8098q^{-1} + 0.8432q^{-2}}.$$

First of all, the CRHPC and CRHPC$_1$ have been tried using the same set of tuning parameters as the previous section, namely $[N, m, \mu(j), \rho(j)] = [4, 3, 1, 1]$, which satisfy the conditions of Theorems 2.1 and 2.5. The results are shown in Fig.2.3 for the 2-norm controller and in Fig.2.4 for the 1-norm counterpart. As expected, both of them are stable and the optimal cost function values are non-increasing. The difference between the cost function values of the 2-norm and the 1-norm cases is a consequence of the use



(a) Input/Ouput responses for the CRHPC$_1$

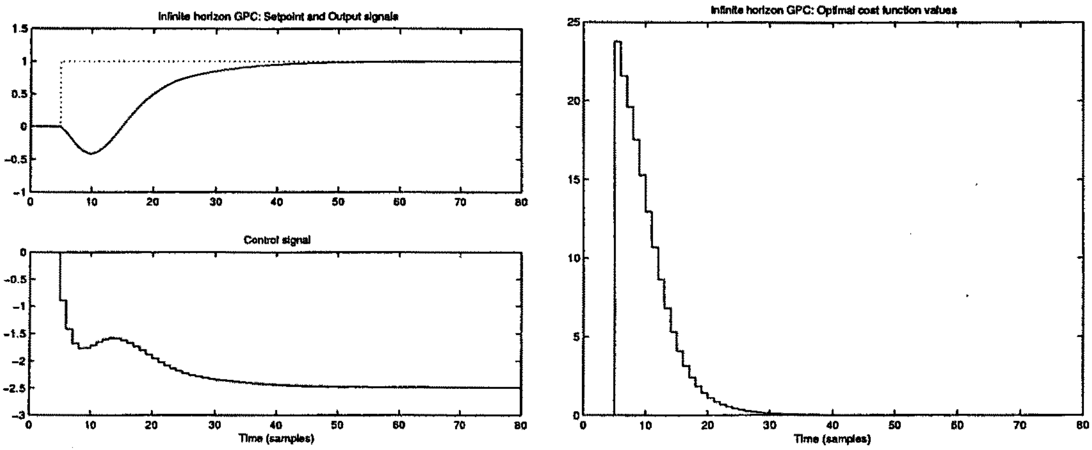
(b) Optimal cost function values for the CRHPC$_1$

Figure 2.4: Closed-loop behaviour of the CRHPC$_1$: $[N, m, \mu, \rho] = [4, 3, 1, 1]$

of the square in the latter. The input/output closed-loop responses, which are quite similar for both controllers, exhibit a deadbeat-like behaviour with a considerable initial inverse response.

For this example, the CRHPC and the GPC$^\infty$ are not equivalent, since the system is not open-loop antistable. The GPC$^\infty$ and the GPC$_1^\infty$ have been tried with $[N_u, \rho(j)] = [3, 1]$. With these settings, the GPC$^\infty$ (for either norm) uses just one end-point equality constraint, as the open-loop system is stable ($n_{\bar{a}}+1 = 1$). Hence, two degrees of freedom are used to attain the minimisation of the cost function, and the other one is needed to enforce the equality constraint. Notice that, in the already presented CRHPC, $m = 3$ and $N_u = 5$, thus the degrees of freedom available for minimisation in those controllers are also $N_u - m = 2$. The GPC$_1^\infty$ iterative algorithm provided in Section 2.3.2 converges for $[n + 1] = 80$ or $[n + 1] = 160$ in this example, depending on the sample.

Fig.2.5 shows the closed-loop behaviour of the GPC$^\infty$ and Fig.2.6 that of the GPC$_1^\infty$. A simulation time of 80 samples (50 more than for the CRHPC examples) have been chosen taking into account the closed-loop dynamics. It is worth pointing out that the



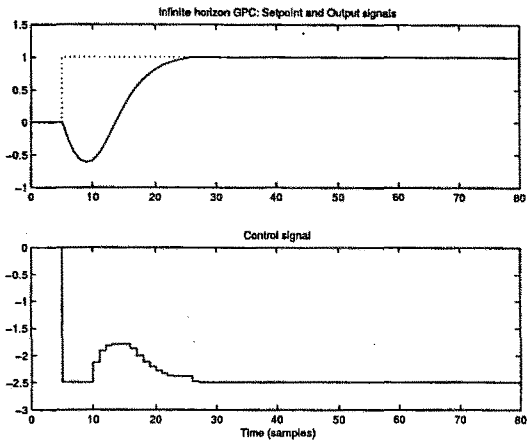
(a) Input/Ouput responses for the (2-norm) GPC$^\infty$

(b) Optimal cost function values for the (2-norm) GPC$^\infty$

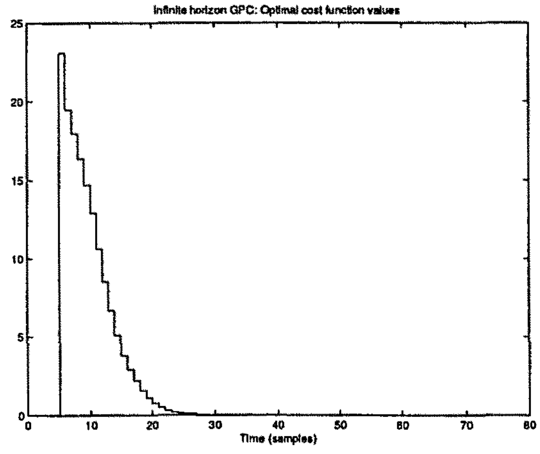
Figure 2.5: Closed-loop behaviour of the (2-norm) GPC$^\infty$: $[N_u, \rho] = [3, 1]$

1-norm and 2-norm responses are quite similar. The output response obtained with the GPC$_1^\infty$ is a bit faster, and the cost function settles down to zero a few samples before than the 2-norm counterpart. In addition, notice that the output shape is softer and the control efforts are lower compared to the CRHPC. As shown in the following chapter, the smoother behaviour of the GPC$^\infty$ relative to the CRHPC leads to better robustness properties.

Finally the QGPC$_1^\infty$ has been tried with the same tuning knobs, *i.e.* $[N_u, \rho(j)] = [3, 1]$. As shown in Fig.2.7, the use of the QGPC$_1^\infty$ tuned in this fashion results in an unstable closed-loop system. Such a behaviour is a consequence of the properties of the open-loop system. The non-minimum phase characteristic combined with a short sampling time leads to an open-loop inverse response which takes 10 samples, as shown in Fig.A.2. Notice that, with these tuning knobs, the prediction horizon (eqn.2.14 is $N = \max\{N_u + n_b - 1, n_{\tilde{a}}\} = \max\{4, 2\} = 4$, which means that all the predictions are made within the inverse response region. This is not relevant with the "exact" GPC$_1^\infty$ implementation of Section 2.3.2, but the upper-bound solution provided by the QGPC$_1^\infty$ does not suffice to stabilise the system. Instability with the QGPC$_1^\infty$ does



(a) Input/Ouput responses for the GPC_1^∞



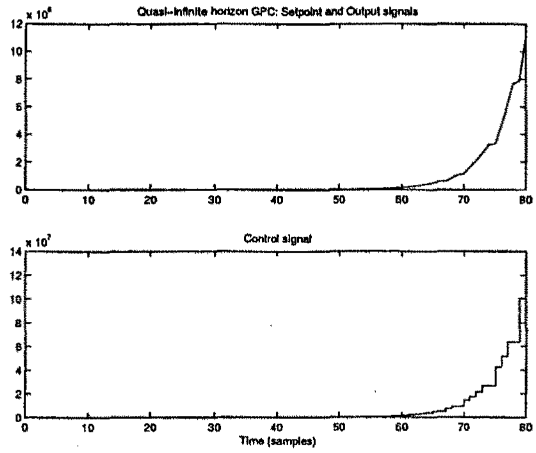
(b) Optimal cost function values for the GPC_1^∞

Figure 2.6: Closed-loop behaviour of the GPC_1^∞: $[N_u, \rho] = [3, 1]$

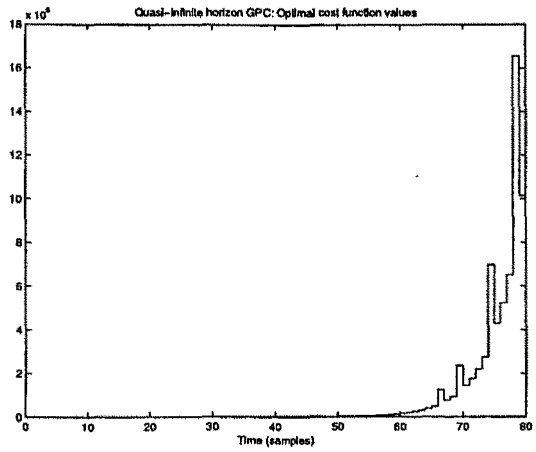
not occur when the system's zero is moved inside the unit circle or when the sampling time is increased in such a way that the initial inverse response takes fewer samples.

The same experiment has been repeated with greater values of the control horizon, and it has been found that the closed-loop system is stable for all $N_u > 4$. It is worth pointing out that N increases with the control horizon as apparent in eqn.2.14, and thus a greater N_u entails a greater N. Fig.2.8 shows the closed-loop behaviour for $[N_u, \rho(j)] = [5, 1]$ attained with the GPC_1^∞ and the $QGPC_1^\infty$. Although the behaviour obtained with the latter is deadbeat-like, the closed-loop system is stable and the cost function is non-increasing. On the other hand, the GPC_1^∞ provides with a smoother response and a less active input signal.

Remark 2.31 Notice that the optimal values of the cost $QGPC_1^\infty$ cost function are lower than those of the GPC_1^∞ for several samples. This is *by no means* a contradiction of Theorem 2.3, since the internal closed-loop states are different for both controllers after the setpoint change, as the computed control sequences are not the same. Theorem 2.3 applies when the infinite horizon, the lower bound and the upper bound problems



(a) Input/Ouput responses for the $QGPC_1^\infty$



(b) Optimal cost function values for the $QGPC_1^\infty$

Figure 2.7: Closed-loop behaviour of the $QGPC_1^\infty$: $[N_u, \rho] = [3, 1]$

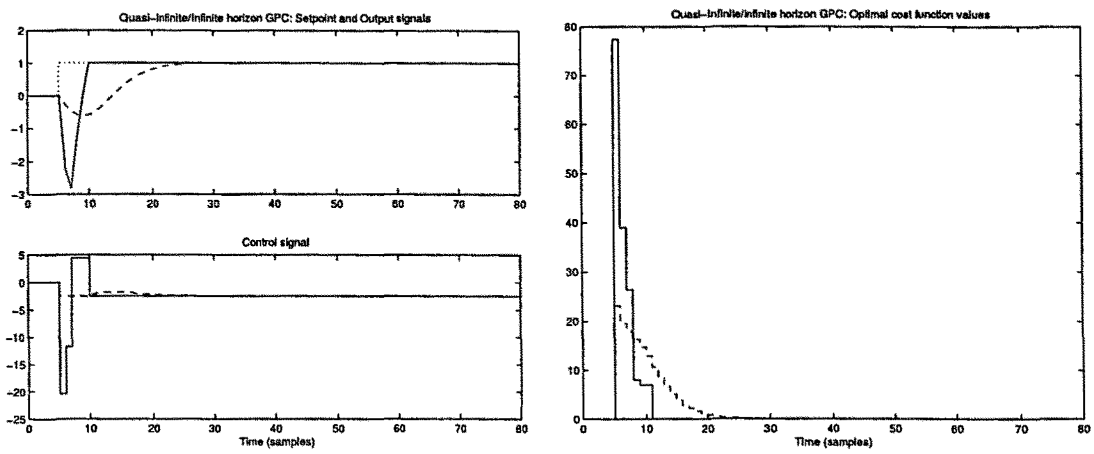
are solved using the same data, *i.e.*, with the same internal states. □□□

Remark 2.32 Not surprisingly, the deadbeat-like behaviour obtained with the GPC_1^∞ closely resembles that obtained with the $CRHPC_1$ (Fig.2.4). The reason for such a situation can be found by comparing the $QGPC_1^\infty$ and the $CHRPC_1$ problems. In the cost function of eqn.2.48 the weight α turns out to be 84.2724 for this particular case. Hence α is about two orders of magnitude greater than the weight on the errors $e(t+j|t)$ for $j = 1, 2, \ldots, N - n_{\bar{a}}$, which is just 1. This means that the term

$$\alpha \sum_{j=N-n_{\bar{a}}+1}^{N} |e(t+j|t)|$$

can be thought of as $n_{\bar{a}}$ *softened constraints* $|e(t+j|t)| \ll 1$, for $j = 1, 2, \ldots, N - n_{\bar{a}}$. These, combined with the $n_{\tilde{a}}+1$ equality constraints on the unstable part of the output, sum up $n_{\bar{a}} + n_{\tilde{a}} + 1 = n_a + 1$ end-point equality constraints, just the same as for the $CRHPC_1$. Hence, for this particular example, the $QGPC_1^\infty$ is closer to the $CRHPC_1$ than to the GPC_1^∞.

This situation is caused by the proximity of the open-loop poles to the unit circle,



(a) Input/Ouput responses for the $QGPC_1^\infty$ (solid) and the GPC^∞ (dashed)

(b) Optimal cost function values for the $QGPC_1^\infty$ (solid) and the GPC^∞ (dashed)

Figure 2.8: Closed-loop behaviour of the $QGPC_1^\infty$ and the GPC^∞: $[N_u, \rho] = [5, 1]$

which makes the powers Φ^j decay to zero very slowly, and thus the upper bound

$$\sum_{j=0}^{\infty} |e(t+N+j|t)| \leq \alpha \sum_{j=N-n_{\tilde{a}}+1}^{N} |e(t+j|t)| \,,$$

is far from being an equality. In the typical case, for classical choices of the sampling time, the maximum absolute value of the open-loop poles is near 0.7, and then the powers Φ^j decay to zero faster, and the bound provided in the above equation is much closer to the equality. As a consequence of this, the $QGPC_1^\infty$ would be closer to the GPC_1^∞ and no deadbeat-like close-loop behaviour is expected. □□□

Section 2.5 shows that the difference between the $QGPC_1^\infty$ and the GPC_1^∞, for this particular example, decreases if greater values of the control horizon are chosen.

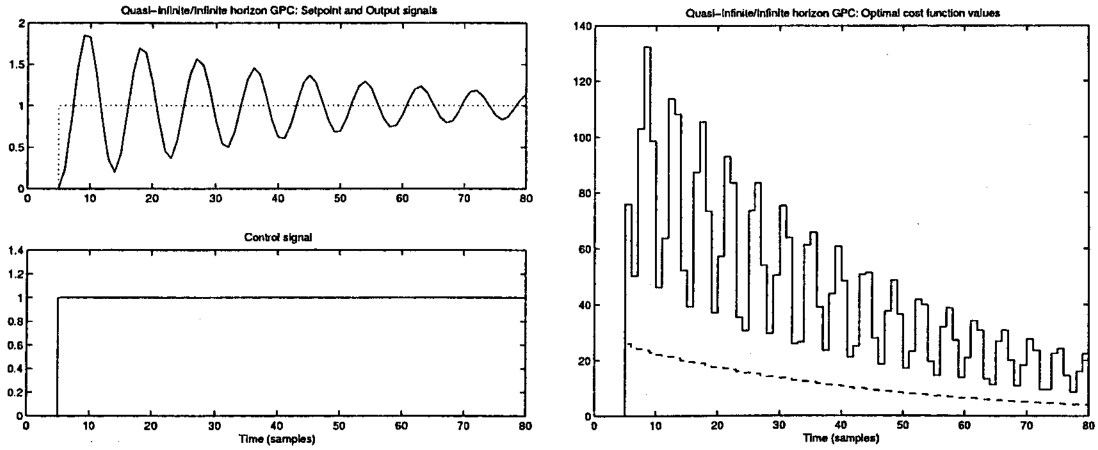
2.4.3 A comparative study of GPC_1^∞ and $QGPC_1^\infty$

This section illustrates that the stability problems of the $QGPC_1^\infty$ discussed in the last section are unusual, and compares the closed-loop behaviour obtained with the GPC_1^∞ and the $QGPC_1^\infty$ for different choices of the control horizon.

Both controllers have been tested against the lightly damped second-order system of eqn.A.10 (Section A.5):

$$G(q^{-1}) = \frac{q^{-1}B(q^{-1})}{A(q^{-1})} = \frac{0.2358q^{-1} + 0.2319q^{-2}}{1 - 1.4835q^{-1} + 0.9512q^{-2}}.$$

Notice that stability with the $\mathrm{QGPC}_{\mathrm{I}}^{\infty}$ is only guaranteed for $N_u = n_{\tilde{a}}+1 = 1$, since Theorem 2.7 applies in that case, however. For $N_u > 1$ the stability of the $\mathrm{QGPC}_{\mathrm{I}}^{\infty}$ must be checked by simulation.



(a) Input/Ouput responses for the $\mathrm{QGPC}_{\mathrm{I}}^{\infty}$ and the $\mathrm{GPC}_{\mathrm{I}}^{\infty}$ (identical)

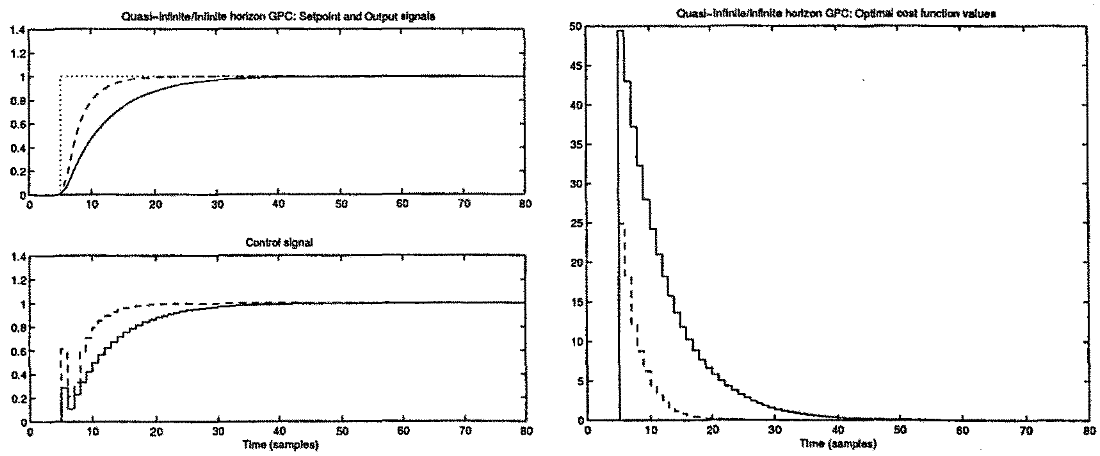
(b) Optimal cost function values for the $\mathrm{QGPC}_{\mathrm{I}}^{\infty}$ (solid) and the $\mathrm{GPC}_{\mathrm{I}}^{\infty}$ (dashed)

Figure 2.9: Closed-loop behaviour of the $\mathrm{QGPC}_{\mathrm{I}}^{\infty}$ and the GPC^{∞}: $[N_u, \rho] = [1, 1]$

With the choice $[N_u, \rho(j)] = [1, 1]$, the $\mathrm{QGPC}_{\mathrm{I}}^{\infty}$ and the $\mathrm{GPC}_{\mathrm{I}}^{\infty}$ are identical, as proved in Theorem 2.7 and shown in Fig.2.9. This choice of N_u leads to mean-level control on the stable part of the process, *i.e.*, the whole process in this case, as discussed in (Scokaert, 1997). Although the input/output responses are the same due to the fact that the only degree of freedom of the controller is used to enforce the end-point equality constraint, the optimal cost function values, plotted in Fig.2.9(b), are not the same. The cost function values of the $\mathrm{GPC}_{\mathrm{I}}^{\infty}$ are a non-increasing sequence, as proved in Theorem 2.6, but the upper bound cost function used in the $\mathrm{QGPC}_{\mathrm{I}}^{\infty}$ does not guarantee that property, as evident in the figure.

Remark 2.33 The condition that the optimal cost sequence is monotonically non-increasing is sufficient but not necessary for stability. The closed-loop system can be stable even though this condition is not satisfied, as occurs for this example. □□□

In addition, notice that a comparison between the sequences of optimal cost function values make sense *for all samples* in this case, since the control efforts computed by both algorithms are the same, and hence the cost function minimised by the QGPC$_1^\infty$ is an upper bound of that of the GPC$_1^\infty$ at all samples. The fact the optimal cost function value computed with the QGPC$_1^\infty$ is always greater than or equal to that of the GPC$^\infty$ is a consequence of Theorem 2.3. This situation is general when $N_u = n_{\tilde{a}} + 1$.



(a) Input/Ouput responses for the QGPC$_1^\infty$ (solid) and the GPC$_1^\infty$ (dashed)

(b) Optimal cost function values for the QGPC$_1^\infty$ (solid) and the GPC$_1^\infty$ (dashed)

Figure 2.10: Closed-loop behaviour of the QGPC$_1^\infty$ and the GPC$^\infty$: $[N_u, \rho] = [2, 1]$

The behaviour obtained with the QGPC$_1^\infty$ and the GPC$_1^\infty$ when the tuning knobs are chosen as $[N_u, \rho(j)] = [2, 1]$ is shown in Fig.2.10. The results do not differ too much, but the response is somewhat faster in the case of the GPC$_1^\infty$. On the other hand, the optimal values of the QGPC$_1^\infty$ cost function are about twice larger than those of the GPC$^\infty$ for the first few samples. Although both closed-loop systems have turned out to be similar, it must be taken into account that the GPC$_1^\infty$ algorithms involves an enormous computational burden, since the convergence condition is achieved for $[n+1] = 360$ at most samples. On the other hand, the QGPC$_1^\infty$ solution is several orders of magnitude faster.

Apart from the graphical comparison provided above, the closed-loop behaviour obtained with both controllers can be compared numerically. Let $u^\infty(t)$ and $y^\infty(t)$ denote the input/output responses obtained with the GPC$_1^\infty$, and $u^{Q\infty}(t)$ and $y^{Q\infty}(t)$ those obtained with the QGPC$_1^\infty$. The "distance" between both solutions can be computed using the criterion

$$J_{\text{dif}} = \begin{cases} \sum_{t=1}^{n_t} \left[y^\infty(t) - y^{Q\infty}(t)\right]^2 + \phi\left[u^\infty(t) - u^{Q\infty}(t)\right]^2, & \text{stable QGPC}_1^\infty, \\ \infty, & \text{otherwise,} \end{cases} \tag{2.49}$$

where n_t is the number of samples of the experiment and ϕ stands for some non-negative scalar. In the last experiment the value obtained for J_{dif} is 1.6926, for $\phi = 1$ and $n_t = 80$. If the experiment is repeated for $N_u > 2$, the difference between the GPC$^\infty$ and the QGPC$_1^\infty$ vanishes, *i.e.* $J_{\text{dif}} = 0$ (actually $J_{\text{dif}} < 10^{-27} \approx 0$), with the same choices for ϕ and n_t. Hence, the QGPC$_1^\infty$ and the GPC$_1^\infty$ seem to converge for large enough N_u. The following section is focused on providing a formal framework to justify this hypothesis.

		Control horizon (N_u)				
		1	2	3	4	5
Controller	QGPC$_1^\infty$ ($t_{\text{QGPC}_1^\infty}$)	1.00	1.15	1.07	1.12	1.20
	GPC$_1^\infty$ ($t_{\text{GPC}_1^\infty}$)	13342.42	2489.23	3.08	3.00	3.37
Ratio ($t_{\text{GPC}_1^\infty}/t_{\text{QGPC}_1^\infty}$)		13342.42	2160.26	2.89	2.64	2.80

Table 2.1: Normalised CPU time

Apart from stability and performance, the GPC$_1^\infty$ and the QGPC$_1^\infty$ can be compared in terms of computation requirements. Table 2.1 shows the *Central Processing Unit* (CPU) time[10] required by the 80-sample simulations performed with both con-

[10] *Floating point operations* (flops) are not always counted in Matlab, and hence CPU time has been preferred as a measurement.

trollers. Of course, CPU time varies between different machines, and thus the results are displayed normalised, dividing by the lowest value, in such a way that they provide machine-independent information. There may still be small variations from one machine to other. The results show that the GPC_1^∞ may need much more computational effort than the $QGPC_1^\infty$, even more than 13000 times (!) the CPU time required by the latter (for $N_u = 1$), and always more than twice. For $N_u = 1$ the whole simulation takes less than a second with the $QGPC_1^\infty$, whereas the GPC_1^∞ takes more than two hours (!). This result is quite consistent since, in the best case, the GPC_1^∞ solves two LP problems to compute a single control move (if only one iteration is required for convergence), whereas the $QGPC_1^\infty$ solves just one. Notice, in addition, that although the $QGPC_1^\infty$ and the GPC_1^∞ are identical for $N_u > 2$ (for these particular conditions), the former involves less than half the computational burden of the latter. This is of particular relevance if min-max methods are to be used to robustify these controllers (see Chapter 4).

2.5 From quasi-infinite to infinite horizons

The experiment of Section 2.4.3 for $N_u > 2$ reveals that the $QGPC_1^\infty$ converges to the GPC^∞ to some extent. The scope of this section is to justify that convergence and to show how it can be exploited.

The convergence of the $QGPC_1^\infty$ and the GPC_1^∞ for large enough N_u is due to the combination of two different facts. Firstly, it must be noticed that $N_u \to \infty \Rightarrow N \to \infty$, and hence the cost function of the $QGPC_1^\infty$ (eqn.2.48) converges to that of the GPC_1^∞ (eqn.2.36). However, this would only explain the convergence for very large N_u, and $N_u = 3$ in Section 2.4.3 does not appear to be large enough, especially as the results obtained using the GPC_1^∞ and the $QGPC_1^\infty$ are not as close for $N_u = 2$.

A second reason why a large N_u brings the $QGPC_1^\infty$ closer to the GPC^∞ comes out by paying attention to the upper bound which is used in the cost function of the QGPC_1^∞ (eqn.2.46):

$$\sum_{j=0}^{\infty} |e(t+N+j|t)| \leq \alpha \|z(t)\|_1 = \alpha \sum_{j=N-n_{\bar{a}}+1}^{N} |e(t+j|t)| .$$

If the controllers are provided with enough degrees of freedom for the minimisation of the cost function, then the predicted errors will tend to zero: $|e(t+j|t)| \ll 1$ for large enough j. Thus, if the condition

$$|e(t+j|t)| \approx 0,\ j = N - n_{\bar{a}} + 1, N - n_{\bar{a}} + 2, \ldots, N, \tag{2.50}$$

holds for some N_u, it then follows that

$$0 \leq \sum_{j=0}^{\infty} |e(t+N+j|t)| \leq \alpha \sum_{j=N-n_{\bar{a}}+1}^{N} |e(t+j|t)| \approx 0.$$

Therefore the upper bound (QGPC_1^∞) and the GPC_1^∞ cost functions become almost identical. The larger N_u is, the more likely the condition of eqn.2.50 is to hold since, as more degrees of freedom are available, the controller can lead the output to the setpoint faster. In addition, this condition can be satisfied even for relatively small N_u, what explains why the GPC_1^∞ and the QGPC_1^∞ are identical for $N_u > 2$ in the example mentioned above. Finally, notice that, in such a situation the $\mathrm{QGPC}_1^\infty/\mathrm{GPC}_1^\infty$ would also be quite close to the CRHPC_1, for the reasons pointed out in Remark 2.32.

The following conjecture stems from these observations:

Conjecture 2.1 *For any stabilisable and detectable system, there exists a relatively small integer* $\overline{N}_u > n_{\bar{a}}$ *such that the* QGPC_1^∞ *is stabilising if*

(i) $\rho(N_u) \geq \rho(N_u - 1) \geq \cdots \geq \rho(1) \geq 0$, *and*

(ii) $N_u \geq \overline{N}_u$.

Remark 2.34 As discussed above, the increase of N_u can result on the satisfaction of eqn.2.50, and therefore the distance between the GPC_1^∞ and the QGPC_1^∞ vanishes, leading to stability. However, there is no estimation about how large $\overline{N}_u$ is required for each system. Sometimes the closed-loop system with the QGPC_1^∞ is stable for all $N_u \geq n_{\tilde{a}} + 2$ (the first case which does not satisfy the conditions of Theorem 2.7), as happens in the example shown in Section 2.4.3, whereas some other times larger values may be needed, as shown in the example provided in Section 2.4.2. ❑❑❑

N_u	2,3,4	5	6	7	8	9	10	11	> 11
J_{dif}	∞	572.4	216.0	150.0	89.6	61.7	27.7	19.5	0

Table 2.2: J_{dif} as a function of N_u

The following example is intended to illustrate the convergence property of Conjecture 2.1. A few experiments have been carried with the non-minimum phase system of Section 2.4.2 (eqn.A.4), with $\rho(j) = 1$ for all j and different values of N_u. The distance from the GPC_1^∞ to the QGPC_1^∞ solutions has been computed using the criterion defined in eqn.2.49 with $\phi = 1$ and $n_t = 80$, and the results are displayed in Table 2.2. $N_u = 1$ has not been included in this comparison since Theorem 2.7 applies in that case, leading to $J_{\text{dif}} = 0$. The results show that the QGPC_1^∞ gives rise to an unstable closed-loop system for $1 < N_u < 5$, and the difference between the QGPC_1^∞ and the GPC_1^∞ monotonically decreases from $N_u = 5$ to $N_u = 12$. For $N_u > 11$ the responses obtained with the GPC_1^∞ and the QGPC_1^∞ are identical, since $J_{\text{dif}} = 0$. For this example $\overline{N}_u = 5$, but the QGPC_1^∞ does not provide the same closed-loop behaviour as the GPC_1^∞ until $N_u = 12$.

This example points out that, although the QGPC_1^∞ does not provide with stability guarantees (apart from the cases collected in Theorems 2.8 and 2.7) it is not difficult to tune this controller for stability by increasing the control horizon. This property can be useful when the GPC_1^∞ is to be applied to control a real process, as a great deal of computations can be avoided by using the QGPC_1^∞ formulation instead of the iterative algorithm of Section 2.3.2. Of course, this possibility can be considered only if N_u

is large enough such that the $QGPC_1^{\infty}$ is stabilising (assuming that Conjecture 2.1 is true). In addition, it is worth taking into account that the performance attained with the GPC_1^{∞} and the $QGPC_1^{\infty}$ can be quite different for small N_u, as shown in Section 2.4.2 for $N_u = 5$.

2.6 Concluding remarks

In this chapter several receding-horizon predictive controllers have been formulated. The controllers introduced in Section 2.2, above all the CRHPC and the GPC^{∞}, were proposed to overcome the stability problems of the classical GPC, which were pointed out by Bitmead *et al.* (1990). Both the CRHPC and the GPC^{∞} are modifications of the GPC control law which enjoy the property that the sequence of optimal cost function values is monotonically non-increasing, which implies closed-loop stability as shown by the theorems of Section 2.2.3.

The novel aspects introduced in this chapter are the 1-norm versions of those stabilising approaches, namely the $CRHPC_1$ and the GPC_1^{∞}. The aim of these formulations is to develop efficient robust predictive controllers based on min-max optimisation, which can be solved with standard LP tools if 1-norm cost functions are used, as shown in (Camacho and Bordóns, 1995). The $CRHPC_1$ does not introduce any difficulty, and can be easily solved as a LP problem. On the other hand, the GPC_1^{∞} requires the use of an iterative algorithm until the difference between a lower bound and an upper bound solutions is small enough. Finally, a simple upper bound problem of the GPC_1^{∞}, referred to as $QGPC_1^{\infty}$, is defined in Section 2.3.3. The $QGPC_1^{\infty}$ avoids the computational burden involved by the GPC_1^{∞}, since the iterative algorithm depicted in Section 2.3.2 is not used. However, the stability guarantees with the $QGPC_1^{\infty}$ reduce to a few special cases. Several theorems proving the stability of these 1-norm controllers under some conditions are provided in Section 2.3.4.

In the simulation examples of Section 2.4, it can be observed that both the CRHPC and the GPC^{∞}, either in the 2-norm or the 1-norm formulations, lead to a stable closed-loop system. Apart from stability, the GPC^{∞} seems a better choice as performance is considered, since the CRHPC often produces a deadbeat-like behaviour (especially if a short prediction horizon is chosen, as shown in the forthcoming chapters).

Finally, the convergence of $QGPC_1^{\infty}$ and GPC_1^{∞} is analysed in Section 2.5, where the fact that, for a large enough (but relatively small) control horizon, the closed-loop behaviour provided by these two controllers is indistinguishable is illustrated by means of an example. Hence, although the $QGPC_1^{\infty}$ itself is not always stabilising, it is conjectured that with a large enough N_u it turns to be a stabilising controller since it converges to the GPC_1^{∞}. This property is remarkably useful because it allows to implement the GPC_1^{∞} as the $QGPC_1^{\infty}$ for sufficiently large N_u, avoiding the high computational requirements of the iterative algorithm depicted in Section 2.3.2.

Chapter 3

Robust analysis and design of unconstrained stabilising RHPC

3.1 Introduction

In Chapter 2 some predictive controllers which guarantee the stability of the nominal closed-loop system are presented. These controllers ensure closed-loop stability as far as the system to which they are applied, referred to as *true system/process/plant* hereafter, is identical to the internal model used for predictions, referred to as *nominal model/system/process/plant* throughout this thesis. However, in real applications, the true system and the nominal model always differ, since models, and especially linear representations, are merely an approximation to reality. The sources of uncertainty in the plant model are numerous and the list given below (Skogestad and Postlethwaite, 1996) points out some relevant ones:

1. There are always parameters in the model which are only known approximately or are simply in error.
2. The parameters in the linear model may vary due to non-linearities or changes in the operating conditions.
3. Measurement devices have imperfections. This may even give rise to uncertainty

in the manipulated input, since the actual input is often adjusted in a cascade manner.

4. At high frequencies even the structure and the model order is unknown.
5. Even when a very detailed model is available, a simpler lower order nominal model can be used and the neglected dynamics are represented as uncertainty.
6. The controller implemented may differ from the one obtained by solving the synthesis problem. In this case one may include uncertainty to allow for controller order reduction and implementation inaccuracies.

It is not in the scope of this thesis to provide a full review of the uncertainty sources, but to suggest methods to incorporate some knowledge of uncertainty into the control design. If system uncertainty is overlooked when the control algorithms introduced above are to be used in real applications, the consequences can be dramatic, including instability, constraint violations and poor performance. It must be noted that it would be a worthless and senseless effort to design a controller according to some optimality criterion, such as the minimisation of a cost function, if, after all, the closed-loop behaviour is spoiled by modelling errors and disturbances.

If predictive controllers are used as a way of obtaining an optimal input/output behaviour and constraints are not an important issue, a classical approach to robustness becomes possible. The scope of this chapter is to undertake the robustness analysis and design making use of well-known classical tools, such as the *small gain theorem* (Morari and Zafiriou, 1989; Skogestad and Postlethwaite, 1996). First of all, a few preliminary results are presented to show that 2-norm predictive controllers can be posed in the classical *two-degrees-of-freedom* (2-DOF) LTI configuration. The robustness properties can then be analysed using standard techniques.

In this chapter, the stabilising controllers introduced in Chapter 2 are analysed in terms of robustness. *The target of this survey is to find out the inherent robustness* *properties of different stabilising strategies.* This analysis should provide with some insight on how uncertainty can be handled by predictive controllers. In addition, this inquiry and can be useful to point out candidates for robust constrained predictive controllers, an issue which is undertaken in the following chapters.

The synthesis of robust unconstrained MPC has been addressed from several points of view. Among these, the two most commonly found approaches which use input/output formulations (GPC-like controllers) are the well-known *T-design* and *Q-parametrisation* (or *Q-design*). The former is based on tuning the polynomial T of the nominal model (Clarke and Mohtadi, 1989; Robinson and Clarke, 1991; Soeterboek, 1992; de Prada *et al.*, 1994; Yoon and Clarke, 1995*a*; Megías, 1996; Megías *et al.*, 1996; Megías *et al.*, 1997), and the latter relies on parametrising, via a rational function Q, all the controllers which lead to the same nominal transfer function (Kouvaritakis *et al.*, 1992; Yoon and Clarke, 1995*a*; Hrissagis *et al.*, 1996; De Nicolao *et al.*, 1996; Ansay and Wertz, 1997). The so-called observer polynomial T has been shown to be a valuable tool to enhance robustness since this tuning knob does not affect the nominal closed-loop behaviour. However, the T-design methods are based on heuristic rules, whereas Q is chosen to optimise some robustness criterion. Thus the Q-design methods are systematic, a reason for which these are often preferred. Despite that, the T-design tends to providing greater robustness margins as remarked in (Yoon and Clarke, 1995*a*; Ansay and Wertz, 1997; Megías *et al.*, 1999*a*).

In this chapter, *a systematic procedure to design T—T-optimisation— which is not heuristic but based on optimising a robustness criterion is presented.* The aim of this methodology is to combine the advantages of the heuristic T-design and the systematic Q-parametrisation methods and, at the same time, to overcome their drawbacks. The idea of choosing T by means of optimisation was firstly outlined in (Megías, 1996; Megías *et al.*, 1997) and finally exploited in (Megías *et al.*, 1999*a*).

This chapter is structured in seven sections. In Section 3.2 LTI forms for unconstrained 2-norm RHPC controllers are derived. These allow a classical approach to analyse robustness. The robust design of unconstrained GPC-like controllers through the polynomial T is presented in Section 3.3. Section 3.4 compares the robustness of CRHPC, GPC$^\infty$, and a "softened" CRHPC. As the true plant is never available for simulation, the only criteria used to analyse different controllers are nominal performance and robust stability, although simulations with a *true* plant are provided. Section 3.5 presents the Q-parametrisation method and analyses several choices of the parameter Q suggested in the literature. In Section 3.6, the T-optimisation procedure is defined and compared with other approaches, showing that it can provide with greater robustness bounds. Finally, Section 3.7 draws the most relevant conclusions of this chapter.

3.2 The classical approach to robustness

The classical robustness analysis (and design) of unconstrained RHPC is possible due to the LTI form of these controllers which was proposed in (Bitmead *et al.*, 1990) for the GPC and, later, extended to the CRHPC (Scokaert, 1994; Yoon, 1994; Yoon and Clarke, 1995*a*). Here, these formulae are also obtained for the infinite horizon GPC$^\infty$ of (Scokaert, 1994; Scokaert, 1997).

This approach assumes that *both* the true and the nominal systems are linear and time invariant. The nominal system is given by the CARIMA model of eqn.2.1 in Chapter 2:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{T(q^{-1})}{\Delta}\xi(t),$$

whereas the true process is described by a *different* input/output model:

$$A_0(q^{-1})y(t) = B_0(q^{-1})u(t-1) + x(t),$$

where A_0 and B_0 are *unknown* polynomials on the backward shift operator and $x(t)$ is an additive disturbance at time t. Notice that $x(t)$ affects the internal states, and not only the output $y(t)$. The perturbation $x(t)$ can be written as an additive disturbance at the output $d_y(t)$ (Fig.3.1) since the equivalence

$$d_y(t) = \frac{1}{A_0(q^{-1})} x(t)$$

clearly holds.

3.2.1 Closed-loop formulae of unconstrained RHPC

This section provides with an equivalent 2-DOF LTI form of unconstrained RHPC. The closed-loop formulae presented here are valid only for quadratic cost functions, that is, those which fit the definition of eqn.2.3.

Remark 3.1 Throughout this chapter it is assumed that no future setpoints are available, *i.e.* $w(t+j|t) = w(t|t)$ for all $j \geq 0$. □□□

Prior to obtaining the LTI form of RHPC, the polynomials $E_j(q^{-1})$, $F_j(q^{-1})$, $G_j(q^{-1})$ and $H_j(q^{-1})$ are defined to satisfy the system of Diophantine equations:

$$\begin{aligned} T(q^{-1}) &= A(q^{-1})\Delta E_j(q^{-1}) + q^{-j} F_j(q^{-1}), \\ E_j(q^{-1})B(q^{-1}) &= G_j(q^{-1})T(q^{-1}) + q^{-j} H_j(q^{-1}). \end{aligned} \tag{3.1}$$

The details about these polynomials are extensively documented in the literature (Clarke *et al.*, 1987; Clarke and Mohtadi, 1989; Bitmead *et al.*, 1990; Soeterboek, 1992; Megías, 1996) and hence are omitted here for brevity.

Given the definition of $\boldsymbol{\Delta u}^{\text{opt}}(t)$ provided in eqn.2.12 for the finite horizon case, the first postulated control move can be written as

$$\Delta u^{\text{opt}}(t|t) = \sum_{j=N_1}^{N_2} k_j [w(t+j|t) - f(t+j|t)], \tag{3.2}$$

where the coefficients k_j can be computed from

$$\left[\begin{array}{ccc} k_{N_1} & \dots & k_{N_2} \end{array}\right] =$$

$$\left[\begin{array}{cccc} 1 & 0 & \dots & 0 \end{array}\right] \left[\begin{array}{cc} \boldsymbol{G}_1^{\mathrm{T}}\boldsymbol{M}\boldsymbol{G}_1 + \boldsymbol{R} & \boldsymbol{G}_2^{\mathrm{T}} \\ \boldsymbol{G}_2 & -\frac{\gamma}{\mu(N_y)}\boldsymbol{I} \end{array}\right]^{-1} \left[\begin{array}{cc} \boldsymbol{G}_1^{\mathrm{T}}\boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{array}\right]. \quad (3.3)$$

The form of eqn.3.2 is essential for the obtention of an equivalent LTI form of RHPC controllers.

A few manipulations are required to obtain analogous coefficients k_j for the GPC$^\infty$. The expression provided in eqn.2.29 must be rewritten in such a way that the free response and the setpoint values associated to the unstable part of the model, $\tilde{f}(t+N+j|t)$ and $\tilde{w}(t+N+j|t)$ respectively, are given in terms of $f(t+j|t)$ and $w(t+j|t)$.

First of all, notice that eqn.2.18 implies that

$$\tilde{f}(t+j|t) = f(t+j|t) + \tilde{a}_1 f(t+j-1|t) + \dots + \tilde{a}_{n_{\tilde{a}}} f(t+j-n_{\tilde{a}}|t),$$

hence

$$\left[\begin{array}{c} \tilde{f}(t+N|t) \\ \tilde{f}(t+N+1|t) \\ \vdots \\ \tilde{f}(t+N+n_{\tilde{a}}|t) \end{array}\right] = \boldsymbol{H}_{\tilde{A}} \left[\begin{array}{c} f(t|t) \\ f(t+1|t) \\ \vdots \\ f(t+N+n_{\tilde{a}}|t) \end{array}\right], \quad (3.4)$$

where[1] $f(t|t) = y(t)$ by definition and

$$\boldsymbol{H}_{\tilde{A}} = \left[\begin{array}{ccccccccccc} \overbrace{0 & \dots & 0}^{(N-n_{\tilde{a}})} & \tilde{a}_{n_{\tilde{a}}} & \dots & \tilde{a}_1 & 1 & \overbrace{0 & 0 & \dots & 0}^{(n_{\tilde{a}})} \\ 0 & \dots & 0 & 0 & \tilde{a}_{n_{\tilde{a}}} & \dots & \tilde{a}_1 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 & \tilde{a}_{n_{\tilde{a}}} & \dots & \tilde{a}_1 & 1 & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & 0 & \tilde{a}_{n_{\tilde{a}}} & \dots & \tilde{a}_1 & 1 \end{array}\right].$$

Then, if the vector $\boldsymbol{f}'(t)$ is defined as

$$\boldsymbol{f}'(t) = \left[\begin{array}{cccc} f(t|t) & f(t+1|t) & \dots & f(t+N+n_{\tilde{a}}|t) \end{array}\right]^{\mathrm{T}},$$

[1] $f(t|t)$ may be necessary in case that $N = n_{\tilde{a}}$.

it is possible to rewrite eqn.3.4 in the form

$$\widetilde{\boldsymbol{f}}(t) = \boldsymbol{H}_{\tilde{A}}\boldsymbol{f}'(t),$$

where $\widetilde{\boldsymbol{f}}(t)$ is defined in eqn.2.19. In addition, note that if $w(t+j|t) = w(t|t)$ for all $j \geq 0$, it follows from eqn.2.21 that

$$\begin{bmatrix} \tilde{w}(t+N|t) \\ \tilde{w}(t+N+1|t) \\ \vdots \\ \tilde{w}(t+N+n_{\tilde{a}}|t) \end{bmatrix} = \boldsymbol{H}_{\tilde{A}} \begin{bmatrix} w(t|t) \\ w(t+1|t) \\ \vdots \\ w(t+N+n_{\tilde{a}}|t) \end{bmatrix},$$

or

$$\widetilde{\boldsymbol{w}}(t) = \boldsymbol{H}_{\tilde{A}}\boldsymbol{w}'(t)$$

with

$$\boldsymbol{w}'(t) = \begin{bmatrix} w(t|t) & w(t+1|t) & \dots & w(t+N+n_{\tilde{a}}|t) \end{bmatrix}^{\mathrm{T}}.$$

Thus the tracking errors on the unstable part of the output can be obtained as

$$\widetilde{\boldsymbol{w}}(t) - \widetilde{\boldsymbol{y}}(t) = \boldsymbol{H}_{\tilde{A}}\left[\boldsymbol{w}'(t) - \boldsymbol{f}'(t)\right].$$

Furthermore, the setpoint and free-response vectors $\boldsymbol{w}(t)$ and $\boldsymbol{f}(t)$, as defined in eqn.2.15, can also be obtained from $\boldsymbol{w}'(t)$ and $\boldsymbol{f}'(t)$:

$$\boldsymbol{w}(t) = \begin{bmatrix} \boldsymbol{0}_{N,1} & \boldsymbol{I}_N & \boldsymbol{0}_{N,n_{\tilde{a}}} \end{bmatrix} \boldsymbol{w}'(t),$$
$$\boldsymbol{f}(t) = \begin{bmatrix} \boldsymbol{0}_{N,1} & \boldsymbol{I}_N & \boldsymbol{0}_{N,n_{\tilde{a}}} \end{bmatrix} \boldsymbol{f}'(t),$$

which allows to arrange eqn.2.29 as

$$\begin{bmatrix} \Delta \boldsymbol{u} \\ \boldsymbol{\lambda} \end{bmatrix}^{\mathrm{opt}} = \begin{bmatrix} \boldsymbol{G}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{G} + \boldsymbol{R} & \widetilde{\boldsymbol{G}}^{\mathrm{T}} \\ \widetilde{\boldsymbol{G}} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{0}_{N_u,1} & \boldsymbol{G}^{\mathrm{T}}\boldsymbol{\Lambda} & \boldsymbol{0}_{N_u,n_{\tilde{a}}} \\ & \boldsymbol{H}_{\tilde{A}} & \end{bmatrix} (\boldsymbol{w}' - \boldsymbol{f}').$$

Finally, for the infinite horizon case, the first control move can be written as done in eqn.3.2 for the finite horizon case:

$$\Delta u^{\mathrm{opt}}(t|t) = \sum_{j=0}^{N+n_{\tilde{a}}} k_i[w(t+j|t) - f(t+j|t)], \tag{3.5}$$

where the coefficients k_j are computed as

$$\begin{bmatrix} k_0 & \dots & k_{N+n_{\tilde{a}}} \end{bmatrix} =$$
$$\begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} G^{\mathrm{T}}\Lambda G + R & \widetilde{G}^{\mathrm{T}} \\ \widetilde{G} & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0_{N_u,1} & G^{\mathrm{T}}\Lambda & 0_{N_u,n_{\tilde{a}}} \\ & H_{\tilde{A}} & \end{bmatrix}. \quad (3.6)$$

Now, from eqn.3.2 or 3.5, and going through the steps detailed in (Bitmead *et al.*, 1990) for the GPC, the 2-norm (finite horizon) RHPC and the GPC$^\infty$, introduced in Section 2.2, can be written as the standard polynomial expression

$$R_p(q^{-1})\Delta u(t) = T_p(q^{-1})w(t) - S_p(q^{-1})y(t),$$

for which a block diagram is displayed in Fig.3.1. This scheme represents a classical 2-DOF LTI controller.

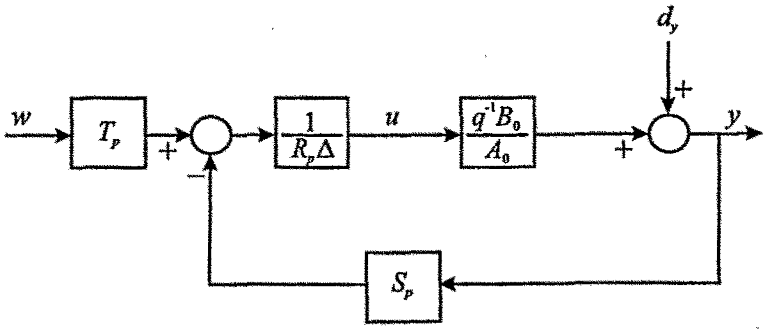


Figure 3.1: 2-DOF structure of RHPC

For the finite horizon case, R_p, S_p and T_p, referred to as 2-DOF polynomials hereafter, can be obtained from the coefficients k_j provided in eqn.3.3 and the polynomials H_j and F_j obtained from the solution of the Diophantine equations of eqn.3.1:

$$\begin{aligned} R_p &= T + q^{-1}\sum_{j=N_1}^{N_2} k_j H_j, \\ S_p &= \sum_{j=N_1}^{N_2} k_j F_j, \\ T_p &= TT_1 = T\sum_{j=N_1}^{N_2} k_j q^j. \end{aligned} \quad (3.7)$$

Notice that if no knowledge about future setpoints is available (as assumed here, in Remark 3.1), the factor T_1 of the polynomial T_p becomes a scalar k_s which multiplies T:

$$T_p = k_s T = \left(\sum_{j=N_1}^{N_2} k_j \right) T.$$

The infinite horizon case is completely analogous to the finite horizon counterpart, and the 2-DOF polynomials can be easily computed from

$$\begin{aligned} R_p &= T + q^{-1} \sum_{j=0}^{N+n_{\tilde{a}}} k_j H_j, \\ S_p &= \sum_{j=0}^{N+n_{\tilde{a}}} k_j F_j, \\ T_p &= T T_1 = T \sum_{j=0}^{N+n_{\tilde{a}}} k_j q^j, \end{aligned} \tag{3.8}$$

with k_j as defined in eqn.3.6. Again, if no preprogrammed setpoints are considered, T_p becomes

$$T_p = k_s T = \left(\sum_{j=0}^{N+n_{\tilde{a}}} k_j \right) T$$

where $T_1 = k_s$ is scalar.

In the 2-DOF LTI block diagram provided in Fig.3.1, the *nominal* closed-loop characteristic polynomial can be obtained taken the true and the nominal systems to be identical or

$$\begin{aligned} B_0 &= B, \\ A_0 &= A, \\ x(t) &= \frac{T}{\Delta} \xi(t). \end{aligned} \tag{3.9}$$

Notice that the second and the third equations imply that the nominal output disturbance d_y takes the form

$$d_y(t) = \frac{T}{A\Delta} \xi(t). \tag{3.10}$$

Now, eqn.3.9 leads to the nominal characteristic polynomial

$$TP_c = AR_p\Delta + q^{-1}BS_p,$$

where the so-called observer polynomial T is a factor what is explicitly written in TP_c. This fact is widely documented in the literature, *e.g.* (Yoon and Clarke, 1995*a*; Megías, 1996), but it must be noticed that T is not seen in the nominal closed-loop transfer functions, since the nominal input/output responses can be obtained as

$$u(t) = \frac{AT_p}{TP_c}w(t) - \frac{AS_p}{TP_c}d_y(t) = \frac{AT_1}{P_c}w(t) - \frac{S_p}{P_c\Delta}\xi(t),$$

and

$$y(t) = \frac{q^{-1}BT_p}{TP_c}w(t) + \frac{AR_p\Delta}{TP_c}d_y(t) = \frac{q^{-1}BT_1}{P_c}w(t) + \frac{R_p}{P_c}\xi(t).$$

For the setpoint tracking problem, only the terms of $u(t)$ and $y(t)$ which involve $w(t)$ are relevant, and the input/output responses come out to be independent of the polynomial T. On the other hand, d_y does not usually fit into eqn.3.10, and hence T determines the speed of disturbance rejection. Thus "slow" dynamics in T should be avoided.

For the many reasons mentioned in Section 3.1, the true and the nominal systems always differ or, in other words, the conditions of eqn.3.9 never hold. Hence, the *true* characteristic equation can be obtained as

$$P_{c0} = A_0R_p\Delta + q^{-1}B_0S_p,$$

where T is no longer a factor. The true input/ouput closed-loop equations are then provided by

$$u(t) = \frac{A_0T_p}{P_{c0}}w(t) - \frac{A_0S_p}{P_{c0}}d_y(t) = \frac{A_0T_p}{P_{c0}}w(t) - \frac{S_p}{P_{c0}}x(t),$$

and

$$y(t) = \frac{q^{-1}B_0T_p}{P_{c0}}w(t) + \frac{A_0R_p\Delta}{P_{c0}}d_y(t) = \frac{q^{-1}B_0T_p}{P_{c0}}w(t) + \frac{R_p\Delta}{P_{c0}}x(t).$$

3.2.2 Uncertainty descriptions

A nominal design is performed using a nominal model G which is assumed to provide with a convenient representation of the true plant G_0. However, the true process cannot be considered as a fixed LTI system due to the reasons pointed out in Section 3.1. Instead of that, G_0 is assumed to lie *somewhere* within a family $\mathcal{F}$ of LTI plants which is defined in terms of G. The formulations provided in this section are focused on SISO systems, but there are analogous results for the MIMO case.

The usual way to specify $\mathcal{F}$ is a frequency domain description which provides with ranges for the true system frequency response, *i.e.* the magnitude and phase for all relevant frequencies. The "true" frequency response is then defined to lie in a region about the nominal frequency response points. In order to reduce the complexity, the uncertainty regions are simplified as a series of discs centred at the nominal points.

In discrete-time formulations, the frequency response of rational functions in q^{-1} is completely determined by the values in the range $0 \leq \omega \leq \pi/T_s$, where T_s is the sampling time and π/T_s is referred to as the Nyquist frequency. Hence, it is quite usual to refer to the *normalised frequency* ω_n, defined as

$$\omega_n = T_s\omega,$$

and then the range $0 \leq \omega_n \leq \pi$ covers all the frequency spectrum.

The family $\mathcal{F}$ of true systems may be described as

$$\mathcal{F} = \left\{G_0(e^{\jmath\omega_n}) : |G_0(e^{\jmath\omega_n}) - G(e^{\jmath\omega_n})| \leq W_a(\omega_n)\right\},$$

or

$$G_0(e^{\jmath\omega_n}) = G(e^{\jmath\omega_n}) + \Delta_a(e^{\jmath\omega_n}),$$

with

$$|\Delta_a(e^{\jmath\omega_n})| \leq W_a(\omega_n).$$

$\Delta_a(e^{\jmath\omega_n})$ is referred to as *additive uncertainty*, since it disturbs the nominal model additively and $W_a(\omega_n)$ stands for the radius of the uncertainty discs.

It is also possible to describe the family $\mathcal{F}$ as

$$\mathcal{F} = \left\{ G_0(e^{\jmath\omega_n}) : \left| \frac{G_0(e^{\jmath\omega_n}) - G(e^{\jmath\omega_n})}{G(e^{\jmath\omega_n})} \right| \leq W_m(\omega_n) \right\},$$

or

$$G_0(e^{\jmath\omega_n}) = [1 + \Delta_m(e^{\jmath\omega_n})]\, G(e^{\jmath\omega_n}),$$

with

$$|\Delta_m(e^{\jmath\omega_n})| \leq W_m(\omega_n).$$

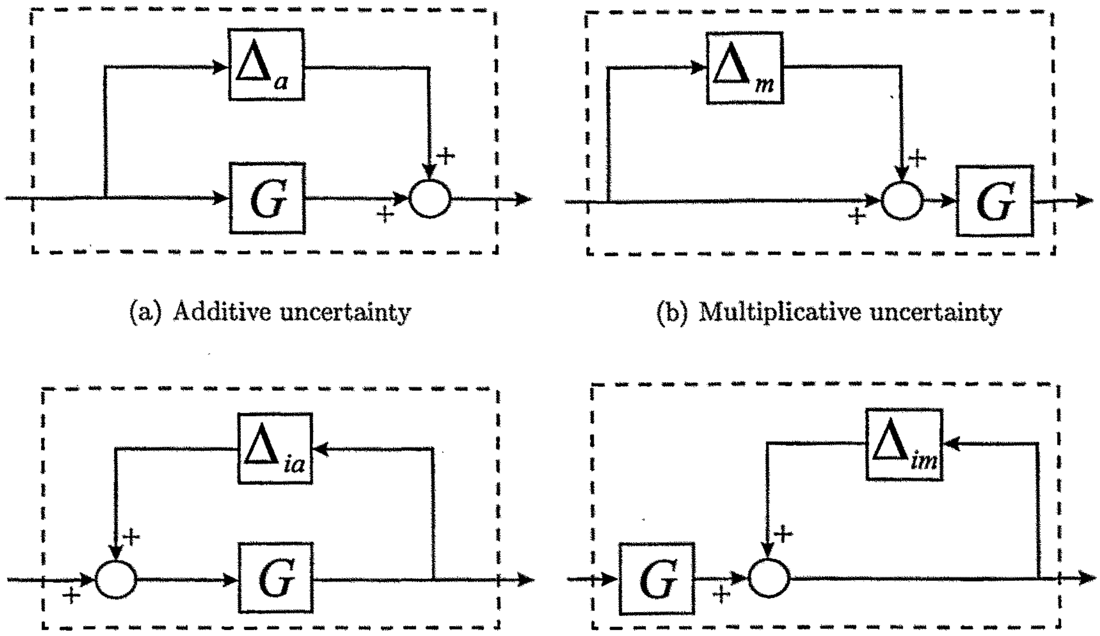
By obvious reasons, $\Delta_m(e^{\jmath\omega_n})$ is referred to as *multiplicative uncertainty*. There exists a simple relationship between additive and multiplicative uncertainties since, by comparing the family and the uncertainty definitions, it follows that

$$W_m(\omega_n) = \frac{W_a(\omega_n)}{|G(e^{\jmath\omega_n})|}, \qquad \Delta_m(e^{\jmath\omega_n}) = \frac{\Delta_a(e^{\jmath\omega_n})}{G(e^{\jmath\omega_n})}.$$

Uncertainty description	**Notation**	**Definition**
Additive	Δ_a	$G_0 = G + \Delta_a$
Multiplicative	Δ_m	$G_0 = G(1 + \Delta_m)$
Inverse additive	Δ_{ia}	$G_0 = (1 - G\Delta_{ia})^{-1}\, G$
Inverse multiplicative	Δ_{im}	$G_0 = (1 - \Delta_{im})^{-1}\, G$

Table 3.1: System uncertainty representations

Apart from these two types of uncertainty, Table 3.1 collects the associated *inverse additive* and *inverse multiplicative* uncertainties. The minus signs in the inverse formulations are not relevant, since uncertainties (additive or multiplicative) are assumed to be bounded *in modulus*. The definitions of the inverse uncertainties are clarified in Fig.3.2, which provides the block diagram for these four types.



(a) Additive uncertainty

(b) Multiplicative uncertainty

(c) Inverse additive uncertainty

(d) Inverse multiplicative uncertainty

Figure 3.2: Uncertainty descriptions

It is worth pointing out that, for MIMO systems, input and output versions of multiplicative uncertainties can be defined. In Fig.3.2 the block diagrams (b) and (d) represent multiplicative input uncertainty and inverse multiplicative output uncertainty respectively. These are simply referred to as multiplicative and inverse multiplicative in Table 3.1, since such a distinction does not apply in the SISO case.

3.2.3 Nominal and robust objectives

When a control system is designed, it is expected that the closed-loop requirements are satisfied for all the possible true plants in the family $\mathcal{F}$. This complex problem is often tackled as two separated tasks:

(1) a nominal design is performed taking into account the nominal model G, and

(2) some uncertainty description is used in order to incorporate the information about

the different dynamics the controller is aimed to manage.

Thus, the control objectives are divided into two different categories:

Nominal objectives: the closed-loop characteristics are considered only with respect to the nominal model G.

Robust objectives: the closed-loop characteristics are considered with respect to the whole family $\mathcal{F}$, which is described in terms of some uncertainty description about the nominal model.

One of the main objectives of most control systems is closed-loop stability. Two different stability problems may be addressed, referred to as *Nominal Stability* (NS) and *Robust Stability* (RS). The former requires that the closed-loop system with the nominal model G is stable, whereas in the latter stability is required for all the plants within the family $\mathcal{F}$. At this point it is worth pointing out that stability is not always a closed-loop requirement. Some systems are expected to work only for a very short time, and then steady-state conditions are not relevant. This happens, for example, in missile control problems. However, in the process industry, stability is a main requirement and must be preserved.

Apart from stability, the control objective is to keep the output $y(t)$ as close as possible to the setpoint $w(t)$ despite input/output disturbances, denoted by $d_u(t)$ and $d_y(t)$ respectively, and measurement noise $n(t)$. This problem is referred to as *Nominal Performance* (NP) when the nominal plant G is taken into account, or *Robust Performance* (RP) when the whole plant family $\mathcal{F}$ is considered.

Disturbance and noise rejection, together with setpoint tracking, is a classical definition of "performance". However, the term "performance", as widely used, often includes a great deal of closed-loop characteristics such as rise time, overshoot, undershoot, input/output variances, control efforts, steady-state errors and so on. A given controller is said to perform "well" or "bad" according to whether these attributes fit the control requirements or not. Nevertheless, the classical disturbance/noise rejection problems must also be taken into account when carrying on with performance measurements. Notice that, in this wider sense, the predictive controllers introduced in Chapter 2 provide with some kind of "optimal performance", since they minimise a multi-objective cost function which involves the tracking errors as well as the control efforts. In addition, the noise model is introduced in eqn.2.1 in such a way that step-like output disturbances are rejected. It is worth pointing out, though, that the end-point equality constraints in the CRHPC (for either norm) introduce some suboptimality in the solution, which can give rise, for instance, to deadbeat-like behaviour, as happens in Section 2.4.2. This kind of poor nominal performance does not usually occur with infinite horizon controllers. Hence, the stabilising controllers of Chapter 2 provide with NS and some degree of NP (especially the infinite horizon approach), but the problem of robustness must be addressed.

In addition, one of the key factors to assess the performance of predictive controllers should be constraint satisfaction since, as mentioned above, constraint handling stands out among the advantages of MPC. This issue is not involved in the unconstrained case analysed here, but should not be overlooked and is taken into consideration in the forthcoming chapters.

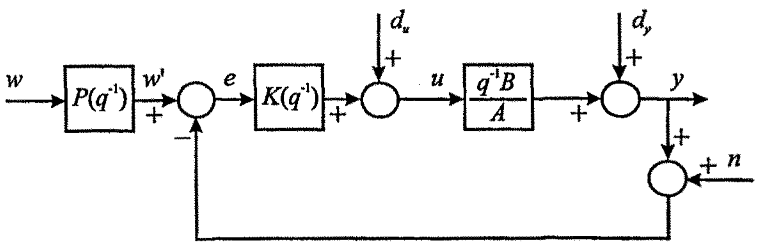


Figure 3.3: RHPC structure with unit feedback

As shown in (Serrano, 1994), the NS, NP, RS and RP problems for unconstrained predictive controllers can be undertaken using standard robust control tools. First of all, it is convenient to rearrange the block diagram of Fig.3.1 to use unit feedback. The result is displayed in Fig.3.3 for the nominal plant $G(q^{-1}) = q^{-1}B/A$, where the *prefilter* $P(q^{-1})$ and the *controller* $K(q^{-1})$ transfer functions can be obtained as

$$P(q^{-1}) = \frac{T_p(q^{-1})}{S_p(q^{-1})}, \qquad K(q^{-1}) = \frac{S_p(q^{-1})}{R_p(q^{-1})\Delta}, \tag{3.11}$$

and it is assumed that the prefilter $P(q^{-1})$ is stable, *i.e.* S_p has no roots outside the unit circle. This assumption does not limit the results presented hereafter, since there is no real necessity of rearranging the block diagram for unit feedback. The analysis presented below can be directly performed on the scheme of Fig.3.1, leading to identical conclusions. However, the unit feedback scheme is often preferred in robust control literature, and hence is used here too.

The steady-state gain of $P(q^{-1})$ is always 1, since $T_p(1) = S_p(1) = k_s T(1)$ as shown, for example, in (Megías, 1996). The error signal $e(t)$ is defined as the difference from the *prefiltered setpoint* $w'(t)$ to the output $y(t)$ in this case. As $w'(t)$ asymptotically converges to $w(t)$, the output is expected to follow the reference, at least for constant setpoints.

A *Zero-Order Hold* (ZOH), which is not explicitly included in Fig.3.3, is assumed in the plant input. In addition, in real applications, an anti-aliasing filter must be used in the feedback path to get rid of any high-frequency component in disturbances or measurement noise, since these cannot be distinguished from low frequency equivalents after sampling. When the output of the plant is only observed after the anti-aliasing filter, and the output values are only considered at the sampling instants, Fig.3.3 provides a simplified digital control structure. It is worth pointing out that, by using discrete-time transfer functions, the plant behaviour between sampling instants is ignored. This can lead to several problems such as the intersampling rippling, which can become very serious.

Sensitivity at the output	*Sensitivity*	$S_y = (I + GK)^{-1}$
	Complementary sensitivity	$T_y = GK(I + GK)^{-1}$
	Control sensitivity	$U_y = K(I + GK)^{-1}$
Sensitivity at the input	*Sensitivity at the input*	$S_u = (I + KG)^{-1}$
	Complementary sensitivity at the input	$T_u = KG(I + KG)^{-1}$
	Control sensitivity at the input	$U_u = G(I + KG)^{-1}$

Table 3.2: Sensitivity transfer functions

Now the robust control theory can be applied making use of the classical sensitivity transfer function definitions:

$$\begin{bmatrix} y \\ u \\ e \end{bmatrix} = \begin{bmatrix} T_y & U_u & S_y & -T_y \\ U_y & S_u & -U_y & -U_y \\ S_y & -U_u & -S_y & -S_y \end{bmatrix} \begin{bmatrix} w' \\ d_u \\ d_y \\ n \end{bmatrix},$$

for the signals shown in Fig.3.3. These closed-loop transfer functions can be obtained as shown in Table 3.2, where G stands for the *nominal* plant. These definitions hold for MIMO systems, and hence I denotes the identity matrix of conformal dimensions. In the SISO case, the identity matrix I can be replaced by 1, and the identities $S_u = S_y$, $T_u = T_y$ and $KU_u = GU_y$ are satisfied.

Remark 3.2 The subindexes $[\cdot]_y$ and $[\cdot]_u$ are introduced for notational clarity, especially for distinction with respect to the noise polynomial T. In robust control literature, the sensitivity functions at the output are often referred to as simply S, T and U, instead of S_y, T_y and U_y, as denoted here. □□□

In the light of Table 3.2, the following binding relationships hold:

$$S_y + T_y = I, \qquad GU_y = T_y,$$
$$S_u + T_u = I, \qquad KU_u = T_u.$$

Thus these sensitivity transfer functions are closely related, what reduces the degrees of freedom available for design.

Now the output $y(t)$ can be expressed in terms of these sensitivity transfer functions and the input signals as

$$y(t) = T_y w'(t) + U_u d_u(t) + S_y d_y(t) - T_y n(t).$$

The NS requirement can be formulated as a classical condition of stability theory. For discrete-time input/output models in the backward shift operator, this means that the roots of closed-loop characteristic polynomial must lie strictly within the unit circle. In unconstrained RHPC, this implies that all the roots of TP_c must be, in modulus, lower than 1.

On the other hand, the NP requirements can be summarised as the condition that the output follows the reference in spite of the disturbances $d_y(t)$ and $d_u(t)$, and the measurement noise $n(t)$, *i.e.*:

$T_y(q^{-1}) \approx I$: to follow the reference $w'(t)$,

$U_u(q^{-1}) \approx 0$: to filter out the input disturbances $d_u(t)$,

$S_y(q^{-1}) \approx 0$: to filter out the output disturbances $d_y(t)$, and

$T_y(q^{-1}) \approx 0$: to filter out the measurement noise $n(t)$.

There seems to be contradiction between the first and fourth objectives $T_y(q^{-1}) \approx I$ and $T_y(q^{-1}) \approx 0$. In addition, in the SISO case, the second objective implies that $K^{-1}T_y(q^{-1}) \approx 0$, which might cause a contradiction with $T_y(q^{-1}) \approx I$ too. However, these requirements can be enforced *at different frequency ranges* with no incompatibility. It is expected that the signals $w'(t)$, $d_u(t)$, $d_y(t)$ and $n(t)$ have different frequency contents, the knowledge of which can be used to design the sensitivity functions at different frequencies.

Notice also that the first and the third requirements are equivalent, since $S_y+T_y = I$. This means that output disturbances with frequency contents in the same range as the prefiltered setpoints are rejected. On the other hand, there is a limitation on the kind of output disturbances and measurement noise which can be filtered out, since it is not possible to enforce $S_y(e^{\jmath\omega_n}) \approx 0$ and $T_y(e^{\jmath\omega_n}) \approx 0$ *at the same frequency* ω_n.

The usual procedure for NP design is to use weighting transfer functions W_S and W_T to specify the desired behaviour of T_y and S_y at different frequencies:

$$S_y W_S \approx 0,$$
$$T_y W_T \approx 0,$$

where W_S and W_T emphasise the appropriate frequency ranges. Hence the NP objectives can be expressed as

$$|S_y(e^{\jmath\omega_n})W_S(e^{\jmath\omega_n})| < 1,$$
$$|T_y(e^{\jmath\omega_n})W_T(e^{\jmath\omega_n})| < 1,$$

for all $0 \leq \omega_n \leq \pi$, which become conditions on the $\mathcal{H}_\infty$ norm of the weighted sensitivity functions:

$$\|S_y(e^{\jmath\omega_n})W_S(e^{\jmath\omega_n})\|_\infty = \max_{0\leq\omega_n\leq\pi} |S_y(e^{\jmath\omega_n})W_S(e^{\jmath\omega_n})| < 1,$$
$$\|T_y(e^{\jmath\omega_n})W_T(e^{\jmath\omega_n})\|_\infty = \max_{0\leq\omega_n\leq\pi} |T_y(e^{\jmath\omega_n})W_T(e^{\jmath\omega_n})| < 1.$$

This method can be directly extended to other sensitivity functions, such as U_u if input disturbances are not negligible.

In (Serrano, 1994, chapter 5), the characteristics of the sensitivity functions S_y and T_y for the GPC are deeply analysed, and the effect of the different tuning knobs on these functions is investigated. This analysis can be readily extended for the unconstrained GPC^∞ and the CRHPC, and analogous conclusions arise. Hence those results about NP are not repeated here. On the other hand, the RS problem is carefully examined in the following section. Finally, the RP objective can be also expressed as conditions on the $\mathcal{H}_\infty$ norm of uncertainty and the sensitivity functions, as discussed in (Serrano, 1994).

In robust control design, it is usually enough to guarantee nominal performance and robust stability, whereas it is accepted that performance is damaged when the true plant is far from the nominal system. Hence, the sequel is mainly concerned with robust stability conditions for unconstrained RHPC, and the robust performance problem is not tackled here.

3.2.4 Robust stability: the small gain theorem

The approach taken to ensure RS is to find a condition such that, provided that the *nominal* closed-loop system is stable (NS), stability is preserved for the whole plant family $\mathcal{F}$. The most useful tool to solve this problem is the *small gain theorem.*

This theorem, as discussed in (Morari and Zafiriou, 1989; Skogestad and Postlethwaite, 1996), can be viewed as an application of the Nyquist stability criterion. Assume that the closed-loop system is stable for the nominal plant G (NS), and that G and G_0 have the same number of unstable poles for the whole family $\mathcal{F}$. Then the true closed-loop system remains stable as far as the Nyquist band of $K(q^{-1})G_0(q^{-1})$ does not include the critical point $-1+0\jmath$. In that case NS implies RS since the number of encirclements to the critical point is the same for the nominal plant G and for all the plants in $\mathcal{F}$. This condition can be formulated as

$$|K(e^{\jmath\omega_n})G_0(e^{\jmath\omega_n}) - K(e^{\jmath\omega_n})G(e^{\jmath\omega_n})| < |1 + G(e^{\jmath\omega_n})K(e^{\jmath\omega_n})|. \tag{3.12}$$

A general MIMO formulation of the small gain theorem is given by below.

Theorem 3.1 (Morari and Zafiriou, 1989; Skogestad and Postlethwaite, 1996) *Assume that M is stable. Then the closed-loop system in Fig.3.4 is stable for all stable perturbations $\|\Delta\|_\infty \leq 1$ if and only if $\|M\|_\infty < 1$.*

Remark 3.3 Alternatively, the stability requirement on Δ may be replaced by the assumption that the number of unstable poles in G and G_0 remains unchanged. ❑❑❑

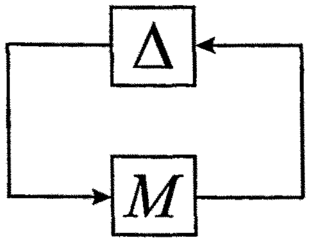


Figure 3.4: General $M - \Delta$ structure for robustness analysis

Proof: See (Morari and Zafiriou, 1989; Skogestad and Postlethwaite, 1996) for a detailed proof of this theorem. ∇∇∇

Remark 3.4 The small gain theorem, as formulated above, requires that the uncertainty Δ is normalised, *i.e.* $\|\Delta\|_\infty \leq 1$. This requirement can be easily obtained for any bounded perturbation. Consider, for example, a bounded additive uncertainty

$$|\Delta_a| \leq W_a \Rightarrow \left|\Delta_a W_a^{-1}\right| \leq 1.$$

Hence, Δ_a can be replaced by the series $\widetilde{\Delta}_a W_a$, where $\widetilde{\Delta}_a = \Delta_a W_a^{-1}$ is a normalised additive uncertainty and W_a is referred to as the uncertainty weight. Obviously, the normalised uncertainty $\widetilde{\Delta}_a$ satisfies $\|\widetilde{\Delta}_a\|_\infty \leq 1$. □□□

Remark 3.5 In the MIMO case, either a couple of input and output weighting matrices

$$\Delta_a = W_a^2 \widetilde{\Delta}_a W_a^1,$$

or a scalar weight

$$\Delta_a = w_a \widetilde{\Delta}_a = \widetilde{\Delta}_a w_a,$$

can be used to achieve the normalisation, as remarked in (Skogestad and Postlethwaite, 1996). □□□

An equivalent formulation of this theorem is the requirement that the closed-loop transfer function from the output $\alpha(t)$ to the input $\beta(t)$ of the (not normalised) uncertainty block times the uncertainty weight is lower than 1 for all frequencies $0 \leq \omega_n \leq \pi$.

In other words, let $\beta(t)$ and $\alpha(t)$ be, respectively, the input and output signals at the uncertainty block Δ, and let V be the closed-loop transfer function from $\alpha(t)$ to $\beta(t)$. Then M, as defined in Theorem 3.1, can be computed as V times the uncertainty weight (W), and the RS requirement is that $\|VW\|_\infty < 1$.

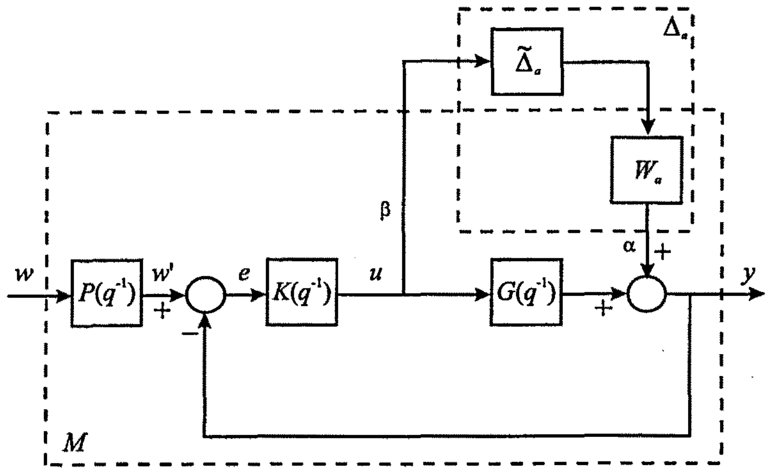


Figure 3.5: Block diagram for additive uncertainty

As an example, the small gain theorem is applied to the additively disturbed case shown in Fig.3.5. V, the transfer function from $\alpha(t)$ to $\beta(t)$, can be obtained as

$$V = -K(1 + GK)^{-1} = -U_y,$$

and hence, $M = VW_a$ leads to the RS condition

$$\|M\|_\infty = \|VW_a\|_\infty = \|-U_yW_a\|_\infty = \|U_yW_a\|_\infty < 1,$$

or

$$|U_y(e^{\jmath\omega_n})W_a(e^{\jmath\omega_n})| < 1 \Leftrightarrow |W_a(e^{\jmath\omega_n})| < \left|\frac{1}{U_y(e^{\jmath\omega_n})}\right|, \tag{3.13}$$

for all $0 \leq \omega_n \leq \pi$.

If the small gain theorem is applied to the different uncertainty descriptions of Table 3.1, the results can be written as conditions on the $\mathcal{H}_\infty$ norm of weighted sensitivity functions. These different RS conditions are collected in Table 3.3.

Description		Condition
Additive	$G_0 = G + \Delta_a$	$\|\|U_y W_a\|\|_\infty < 1$
Multiplicative	$G_0 = G\,(1 + \Delta_m)$	$\|\|T_u W_m\|\|_\infty < 1$
Inverse additive	$G_0 = (1 - G\Delta_{ia})^{-1}\,G$	$\|\|U_u W_{ia}\|\|_\infty < 1$
Inverse multiplicative	$G_0 = (1 - \Delta_{im})^{-1}\,G$	$\|\|S_y W_{im}\|\|_\infty < 1$

Table 3.3: Robust stability conditions

3.2.4.1 The small gain theorem for additive uncertainty

Taking into account the definition of K in eqn.3.11 for RHPC controllers, the sensitivity function U_y (Table 3.2) can be computed as

$$\begin{aligned} U_y &= K(1+GK)^{-1} \\ &= \frac{S_p}{R_p\Delta}\left(1+\frac{q^{-1}B}{A}\frac{S_p}{R_p\Delta}\right)^{-1} \\ &= \frac{AS_p}{AR_p\Delta + q^{-1}BS_p} \\ &= \frac{AS_p}{TP_c}. \end{aligned}$$

Now, the small gain condition of eqn.3.13 can be formulated as done in (Yoon and Clarke, 1995*a*):

Provided that P_c (the nominal characteristic polynomial) is stable, then the real characteristic equation remains stable if A and A_0 have the same number of unstable roots, and if

$$\left|\frac{q^{-1}B_0}{A_0} - \frac{q^{-1}B}{A}\right| < \left|\frac{P_c}{A}\right|\left|\frac{T}{S_p}\right|, \tag{3.14}$$

for all $0 \le \omega_n \le \pi$.

The left-hand side of eqn.3.14 fits the definition of additive uncertainty Δ_a, which can be easily obtained as

$$\begin{aligned}\Delta_a &= G_0 - G\\ &= \frac{q^{-1}B_0}{A_0} - \frac{q^{-1}B}{A}\\ &= \frac{q^{-1}B_0A - q^{-1}BA_0}{A_0A}.\end{aligned}$$

The condition of eqn.3.14 is identical to eqn.3.13 since it can be written as

$$|\Delta_a| < \left|\frac{1}{U_y}\right|,$$

and $|\Delta_a|$ can be replaced by its upper bound $|W_a|$.

The right-hand side of eqn.3.14 can be used to compare the robustness of different controllers. It is easily observed that the polynomial T, which, as already remarked, does not affect the nominal closed-loop transfer functions, can be used to push the robustness bound far from the modelling errors.

3.2.4.2 The small gain theorem for inverse multiplicative uncertainty

The small gain theorem applied for additive uncertainty requires, as a major assumption, that the nominal model and the true system have the same number of unstable poles. This may be a limitation when one or more true poles may cross the unit circle, as sometimes occurs. In such a situation, an inverse uncertainty formulation, *e.g.* inverse multiplicative, can be helpful since a stable uncertainty can be found even when the number of unstable poles varies (Maciejowski, 1989; Skogestad and Postlethwaite, 1996). If a stable uncertainty is available, the small gain theorem (Theorem 3.1) can be applied although G and G_0 do not have the same number of unstable poles.

For GPC-like methods, the small gain theorem for inverse multiplicative uncertainty can be formulated as

Provided that P_c (the nominal characteristic polynomial) is stable, then the *real characteristic equation remains stable if there are no uncertain unstable zeroes and if*

$$\left|\frac{q^{-1}B_0/A_0 - q^{-1}B/A}{q^{-1}B_0/A_0}\right| < \left|\frac{P_c}{A}\right|\left|\frac{T}{R_p\Delta}\right|, \tag{3.15}$$

for all $0 \le \omega_n \le \pi$.

The dependence of this robustness bound with the observer polynomial T is obvious, similarly as happens for an additive uncertainty in eqn.3.14.

The left-hand side of eqn.3.15 is the definition of inverse multiplicative uncertainty Δ_{im}, which can be computed as

$$\begin{aligned}\Delta_{im} &= \frac{G_0 - G}{G_0}\\ &= \frac{q^{-1}B_0/A_0 - q^{-1}B/A}{q^{-1}B_0/A_0}\\ &= \frac{\left(q^{-1}B_0A - q^{-1}BA_0\right)/A_0A}{q^{-1}B_0/A_0}\\ &= \frac{B_0A - BA_0}{B_0A}.\end{aligned}$$

The right-hand side of eqn.3.15, which provides the robustness margin, is the inverse of S_y (see Table 3.3) as defined in Table 3.2:

$$\begin{aligned}S_y &= (1 + GK)^{-1}\\ &= \left(1 + \frac{q^{-1}B}{A}\frac{S_p}{R_p\Delta}\right)^{-1}\\ &= \frac{AR_p\Delta}{AR_p\Delta + q^{-1}BS_p}\\ &= \frac{AR_p\Delta}{TP_c}.\end{aligned}$$

Notice that the small gain theorem for inverse multiplicative uncertainty can be applied as far as uncertainty does not affect unstable zeros. However, in the most general case, uncertainty may alter unstable poles and unstable zeroes. Such a situation can be treated using coprime plant factorisations, as shown in (Skogestad and

Postlethwaite, 1996). This possibility has not been analysed in this thesis for two main reasons:

1. The main aim of this research is to achieve robust constrained predictive controllers. The unconstrained case is used only as a first step to deepen the insight of the problem.
2. If uncertainty is such that it is not possible to be confident, at least, about the number of unstable poles or the number of unstable zeroes of the plant, then MPC may not be a convenient choice, since the internal model is a key factor for the controller behaviour.

3.3 T-design — A heuristic approach

The use of the so-called noise or observer polynomial T as a robustness-enhancing tool is widely documented in the literature (Robinson and Clarke, 1991; Soeterboek, 1992; Yoon and Clarke, 1995*a*; de Prada *et al.*, 1994; Serrano, 1994; Megías, 1996; Megías *et al.*, 1996; Megías *et al.*, 1997).

In this approach, an additive uncertainty representation is often used and then the robustness bound provided in eqn.3.14 applies. Since A and P_c are independent of T, the main aim of this method is to shape the filter S_p/T, which directly appears in the robustness margin. Notice that S_p is a T-dependent polynomial, since it is computed as per eqn.3.7 for the CRHPC and the GPC or eqn.3.8 for the GPC$^\infty$, and hence it depends on the solution of the Diophantine equations (eqn.3.1) which involve T. It can be easily shown that the steady-state gain of this filter is:

$$\frac{S_p(1)}{T(1)} = \sum_j k_j,$$

with $N_1 \leq j \leq N_2$ for the GPC and the CRHPC, and with $0 \leq j \leq N + n_{\tilde{a}}$ for the GPC$^\infty$. This is true regardless the choice of T. Thus the robustness bound at low frequencies cannot be modified. However, the polynomial T can be used to design S_p/T in order to obtain convenient robustness margins at the frequencies for which the modelling errors are greater.

The usual way to choose T is to include the stable part of the model denominator as a factor, or

$$T(q^{-1}) = \tilde{A}(q^{-1})T^*(q^{-1}). \tag{3.16}$$

For stable systems ($A = \tilde{A}$) the factor T^* is often chosen as

$$T^* = (1 - \nu q^{-1})^{n_{t^*}},$$

for some $0 \leq \nu < 1$ and $n_{t^*} \leq N_1$. This choice is known to provide with a simple form to the filter S_p/T.

For stable systems, the easiest possible choice of T in the form of eqn.3.16 is $T = A(1 - \nu q^{-1})$, which is a classical tuning rule suggested, for example, in (Robinson and Clarke, 1991; Soeterboek, 1992; Yoon and Clarke, 1995*a*; Megías, 1996). With this selection, S_p/T becomes a simple low-pass first order filter, as discussed in (Megías, 1996; Megías *et al.*, 1996; Megías *et al.*, 1997). The pole ν can then be used to place the -20 dB per decade slope in the magnitude of S_p/T at lower frequencies. The closer ν is to 1, the lower the bandwidth of S_p/T becomes, what helps to adjust the robustness bound as wished. It is even possible to achieve an infinite robustness bound for stable systems, by choosing $T = A\Delta$. However such a choice is not appropriate since it cancels out integral action, as remarked in (Megías, 1996). In fact, any root which is too close to 1 can cause sluggish disturbance rejection (Robinson and Clarke, 1991) and should be avoided.

This tuning rule is referred to as "heuristic" since, although it was suggested after careful attention on the properties of the filter S_p/T, the reasoning behind this choice is basically qualitative and the results depend on the problem. In addition, the weighting effect of P_c and A on the robustness bound of eqn.3.14 is not taken into account. On the other hand, the Q-parametrisation method used in (Kouvaritakis *et al.*, 1992; Hrissagis *et al.*, 1996; Ansay and Wertz, 1997) is systematic but, as analysed in Section 3.5, does not guarantee better results compared to the T-based schemes. Actually, the robustness bound achieved by using the T-based methods are often greater than those obtained with the Q-design counterparts.

Notice, in addition, that different rules might arise if a different uncertainty representation (*e.g.* inverse multiplicative) were chosen, since the sensitivity function involved in the expression of the robustness bound would differ, as shown in Table 3.3. For inverse multiplicative uncertainty, the relevant filter would be $R_p\Delta/T$ instead of S_p/T.

In this thesis, the robustness bounds provided by eqn.3.14 and 3.15 are exploited to develop a systematic method (Section 3.6), based on optimisation, to increase robustness as much as possible.

3.4 Robust analysis of stabilising RHPC methods

In Chapter 2, some predictive controllers which guarantee NS are provided. The aim of this section to investigate, among those control laws, which should be preferred when uncertainty cannot be overlooked. A heuristic analysis of the robustness features of the CRHPC, the GPC$^\infty$ and a softened version of CRHPC is thus provided.

The uncertain system of Section A.2 has been chosen to proceed with this analysis. A different example which includes a changing number of unstable poles is provided in Section 3.4.3. The real plant is the uncertain third order system defined in eqn.A.3:

$$G_0(q^{-1}) = \frac{q^{-1}B_0(q^{-1})}{A_0(q^{-1})} = (1+\Delta_K)\,\frac{q^{-1}(0.0639 - 0.1110q^{-1} + 0.0437q^{-2})}{1 - 2.6855q^{-1} + 2.4518q^{-2} - 0.7596q^{-3}},$$

whereas the nominal model is the second order system given by eqn.A.4:

$$G(q^{-1}) = \frac{q^{-1}B(q^{-1})}{A(q^{-1})} = \frac{q^{-1}(0.1098 - 0.1232q^{-1})}{1 - 1.8098q^{-1} + 0.8432q^{-2}}.$$

Notice that the true steady-state gain[2] is an uncertain parameter: $K_0 = K(1 + \Delta_K)$, where $K = -0.5$ is the undisturbed true gain and Δ_K is a bounded multiplicative gain uncertainty: $\Delta_K \in [-0.5, 0.5]$.

3.4.1 Nominal design

In this section, the nominal closed-loop behaviour is taken into consideration. Since the polynomial T cancels out in the nominal closed-loop transfer functions from the setpoint $w'(t)$ to the input $u(t)$ and output $y(t)$ (provided that T is stable), $T = 1$ has been used through Sections 3.4.1.1 to 3.4.1.4. This choice reduces the closed-loop characteristic polynomial TP_c simply to P_c.

3.4.1.1 CRHPC with short prediction/control horizons

According to Theorem 2.1, a stable nominal closed-loop system can be obtained with the CRHPC as defined in Section 2.2. For the nominal system of eqn.A.4, the tuning knobs can be taken as $[N \geq 4, m = 3, \rho > 0]$. If the shortest possible prediction horizon $N = 4$ and a constant $\rho(j) = 10^{-2}$ are chosen, this is equivalent to the minimisation of eqn.2.3 with $[N_1, N_y, N_2, N_u, \mu, \rho, \gamma] = [1, 5, 7, 5, 1, 10^{-2}, 0]$. For these tuning knobs, the 2-DOF polynomials computed as per eqn.3.7 become

$$\begin{aligned} R_p(q^{-1}) &= 1 + 60.5868q^{-1}, \\ S_p(q^{-1}) &= -536.1273 + 928.9166q^{-1} - 414.6132q^{-2} \\ &= -536.1273\left[1 - (0.8663 \pm 0.1511\jmath)q^{-1}\right], \\ T_p(q^{-1}) &= -21.8239. \end{aligned} \tag{3.17}$$

[2]The notation K has been chosen for the steady-state gain for distinction with the controller transfer function K.

The closed-loop poles, *i.e.* the roots of the characteristic polynomial P_c, are located at 0.5129 and $0.2860 \pm 0.3033\jmath$, quite inside the unit circle. Hence the resulting closed-loop behaviour is remarkably deadbeat-like, as shown in Fig.3.6.

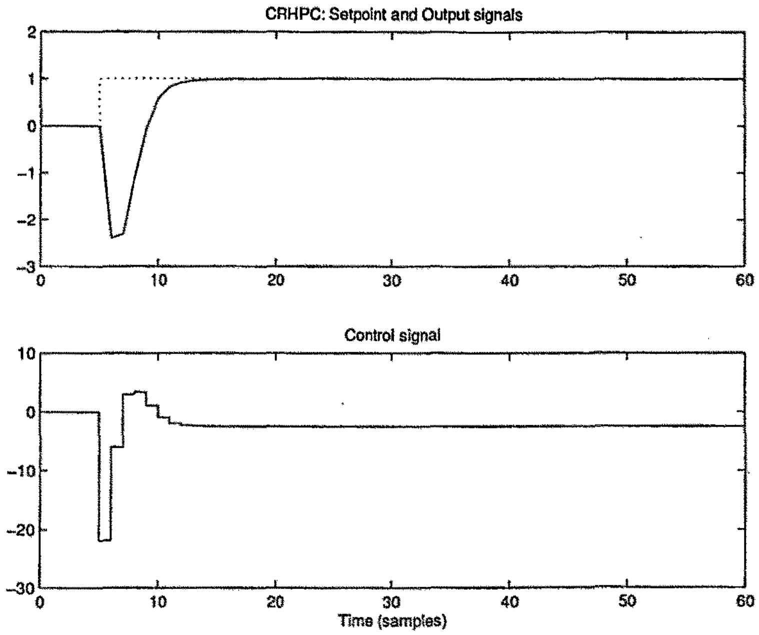


Figure 3.6: Nominal input/output responses for the CRHPC: $[N, m, \rho] = [4, 3, 10^{-2}]$

The reason for this suboptimal deadbeat-like behaviour is a combination of the end-point equality constraints and the use of short prediction and control horizons. As a consequence of this behaviour, very low robustness margins are attained, as shown in Fig.3.9. Two possibilities are analysed in the following two sections so as to reduce this deadbeat-like characteristic.

3.4.1.2 CRHPC with long prediction/control horizons

In this section, the prediction and control horizons are increased. If $N = 25$ is chosen, this results in the tuning knobs $[N_1, N_y, N_2, N_u, \mu, \rho, \gamma] = [1, 26, 28, 26, 1, 10^{-2}, 0]$ in the formulation of eqn.2.3, and the 2-DOF polynomials become

$$\begin{aligned} R_p(q^{-1}) &= 1 + 20.8163q^{-1}, \\ S_p(q^{-1}) &= -177.9097 + 315.8051q^{-1} - 142.4521q^{-2} \\ &= -177.9097\left[1 - (0.8875 \pm 0.1139\jmath)q^{-1}\right], \\ T_p(q^{-1}) &= -4.5566, \end{aligned} \tag{3.18}$$

for which the roots of P_c are 0.8899 and $0.3182 \pm 0.3020\jmath$. The real root is much nearer the unit circle compared to the previous choice of tuning knobs, leading to a much more sluggish output response together with less input activity, as shown in Fig.3.7.

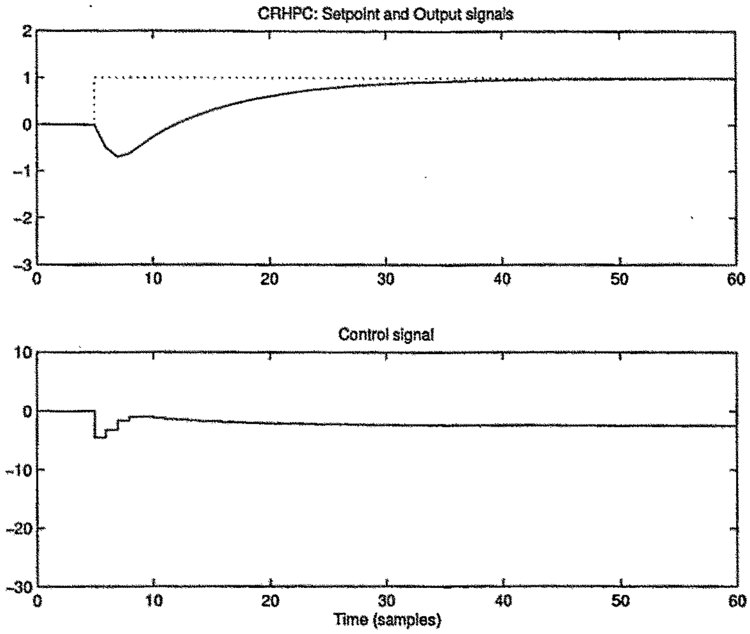


Figure 3.7: Nominal input/output responses for the CRHPC: $[N, m, \rho] = [25, 3, 10^{-2}]$

This result provides with a higher robustness bound with respect to the short horizon choice, as shown in Section 3.4.2. However, this selection of tuning knobs increases the dimension of the problem which is to be solved at each sampling time ($N_u = 26$), what might be relevant if inequality constraints were used, since the QP problem to be solved at each sampling would involve a greater computational burden. In addition, as N_y also increases, numerical instability may arise with the CRHPC, as remarked in (Rossiter and Kouvaritakis, 1994). In such a case, the SGPC would be a more convenient implementation.

3.4.1.3 CRHPC with softened equality constraints

The tuning knob γ in the RHPC formulation of eqn.2.3 can be used to soften the equality constraints giving rise to a less deadbeat-like behaviour. The following experiment has been carried out using the same control/costing/constraint horizons and weights μ and ρ as chosen for the deadbeat-like CRHPC of Section 3.4.1.1, *i.e.* $[N_1, N_y, N_2, N_u, \mu, \rho] = [1, 26, 28, 26, 1, 10^{-2}]$, whereas the parameter γ varies in the interval $[0, 1]$. In other words, the controller is gradually changed from CRHPC ($\gamma = 0$) to GPC ($\gamma = 1$). Fig.3.8 displays the characteristic roots loci obtained with these settings, and it can be observed that whereas the complex roots are almost fixed at their position, the real pole moves to the right and crosses the unit circle. The critical value γ_{lim} for which the nominal closed-loop system becomes unstable (*i.e.* the closed-loop system is stable for all $0 \leq \gamma < \gamma_{\text{lim}}$, and unstable for all $\gamma > \gamma_{\text{lim}}$), has been found at $\gamma_{\text{lim}} = 2.2266 \cdot 10^{-4}$. This analysis, based on the closed-loop pole locations, can be used to choose an suitable value of γ (γ-design).

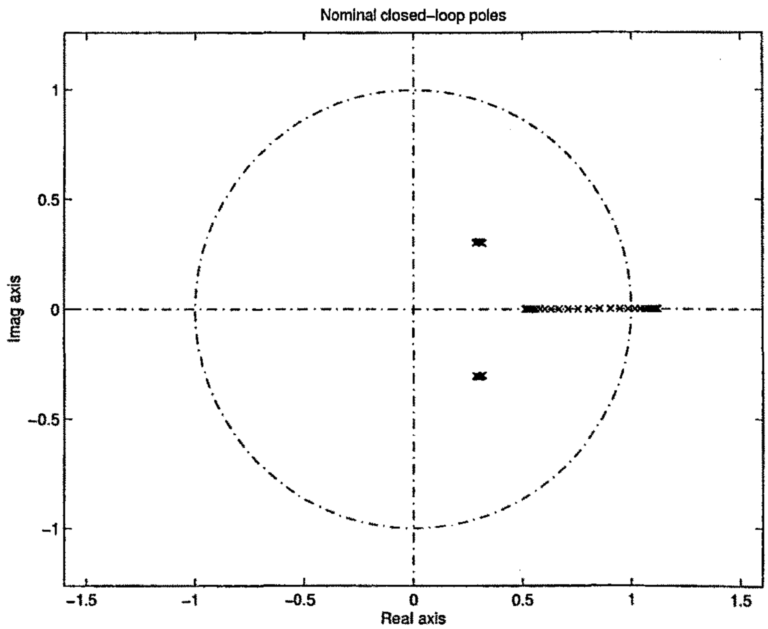


Figure 3.8: Nominal closed-loop poles for the softened CRHPC: $[N, m, \rho] = [4, 3, 10^{-2}]$ and $\gamma \in [0, 1]$

If $\gamma = 10^{-4}$ is chosen, the 2-DOF polynomials for the "softened" CRHPC equal

$$\begin{aligned} R_p(q^{-1}) &= 1 + 19.7053q^{-1}, \\ S_p(q^{-1}) &= -167.8412 + 298.6711q^{-1} - 134.8491q^{-2} \\ &= -167.8412\left[1 - (0.8897 \pm 0.1086\jmath)q^{-1}\right], \\ T_p(q^{-1}) &= -4.0192, \end{aligned} \tag{3.19}$$

for which the roots of P_c are 0.9042 and $0.3139 \pm 0.3043\jmath$, very approximately the same as the ones obtained with the CRHPC presented in the previous section. The 2-DOF polynomials (as the roots are concerned) are quite the same as those of eqn.3.18 especially as compared with those of the deadbeat-like CRHPC in eqn.3.17. Actually, the input/output responses in these two cases are so close that they can be considered indistinguishable, and both controllers can be regarded as *roughly the same* (not equivalent, though).

3.4.1.4 GPC$^\infty$ design

Another way of obtaining a nominally stabilising controller is the GPC$^\infty$ as described in Section 2.2. Since N_1 does not affect stability, $N_1 = 1$ has been taken in this section.

The tuning knobs for the GPC$^\infty$ have been chosen as $[N_u, \rho] = [3, 10^{-2}]$. This choice if consistent with the CRHPC presented in Section 3.4.1.1 from an operational point of view, since both controllers leave two degrees of freedom (control moves) to attain the minimisation of the cost function, and the rest are used to enforce the equality constraints[3]. Although the CRHPC of Section 3.4.1.1 penalises two more control moves (in eqn.2.5) than the GPC$^\infty$ (in eqn.2.6) for these tuning knob choices, the value of ρ is small enough not to take this as a remarkable difference.

[3]It must be pointed out that the GPC$^\infty$ would become a deadbeat law if $N_u = 1$ were chosen, however this is not convenient either for performance or robustness.

The 2-DOF polynomials with this GPC$^\infty$, computed by means of eqn.3.8, are

$$\begin{aligned} R_p(q^{-1}) &= 1 + 20.8695q^{-1}, \\ S_p(q^{-1}) &= -177.8418 + 316.3853q^{-1} - 142.8158q^{-2} \\ &= -177.8418\left[1 - (0.8895 \pm 0.1087\jmath)q^{-1}\right], \\ T_p(q^{-1}) &= -4.2723, \end{aligned} \tag{3.20}$$

and the roots of the characteristic polynomial P_c are located at 0.9035 and $0.2811 \pm 0.2790\jmath$, almost identical to those obtained with the "softened" CRHPC of Section 3.4.1.3 and the CRHPC of Section 3.4.1.2. The 2-DOF polynomials are also very approximately the same as those of eqn.3.18 and 3.19. Hence the input/output responses for this controller are almost superposable to those shown in Fig.3.7.

Surprisingly enough, the softened version of the CRHPC and the GPC$^\infty$ become almost identical for a given value of γ, 10^{-4} in this case. This should not be taken as a sign of controller equivalence, since Chapter 2 clarifies that there are substantial implementation differences between finite and infinite horizon predictive controllers. Only in a few special cases do the CRHPC and the GPC$^\infty$ become identical (see Theorem 2.8).

3.4.1.5 Nominal design — A comparative study

	NS guarantees	Smooth/Deadbeat	Numerical properties
CRHPC (short N)	✔	×	✔
CRHPC (long N)	✔	✔	×
"Softened" CRHPC	×	✔	✔
GPC$^\infty$	✔	✔	✔

Table 3.4: Nominal characteristics of stabilising RHPC

Although the GPC$^\infty$, the "softened" CRHPC and the CRHPC with long horizons have been shown to provide with similar solutions in terms of the 2-DOF polynomials through Sections 3.4.1.2 to 3.4.1.4, there are relevant differences as implementation and the internal properties of these controllers are concerned. These are summarised in Table 3.4.

The use of CRHPC with short prediction/control horizons (such as $N = 4$ in the example), although it guarantees NS, often leads to a suboptimal deadbeat-like closed-loop behaviour. This problem can be overcome in two different ways. On the one hand, the prediction horizon can be increased ($N = 25$ in the example), but this possibility may lead to numerical problems (Rossiter and Kouvaritakis, 1994). In such a case, the theoretically equivalent SGPC (Kouvaritakis *et al.*, 1992) might be used, since it provides with better numerical properties. However, the SGPC solution when constraints are considered is a bit involved, though it is claimed in (Rossiter and Kouvaritakis, 1993) that the constrained SGPC requires much less computational burden compared to the QP techniques used in GPC-like controllers. On the other hand, the equality constraints in the CRHPC can be softened ($\gamma = 10^{-4}$ in the example), providing with a smooth closed-loop behaviour. With such a choice, NS is not guaranteed and must be checked, for example, using a pole-location approach and a suitable value of γ must be found:γ-design.

Finally, among the tested controllers, the GPC$^\infty$ appears to be best choice, since it guarantees the stability of the nominal closed-loop system, it provides smooth behaviour even for short N_u (as far as N_u is chosen greater than the minimal value suggested in Theorem 2.2) and no numerical drawbacks are expected.

3.4.2 Robustness analysis

In this section the robustness of the deadbeat-like CRHPC presented in Section 3.4.1.1 is compared to that provided by the GPC$^\infty$ as tuned in Section 3.4.1.4. The robustness properties of the latter apply also to the CRHPC with long prediction/control horizons of Section 3.4.1.1 and the softened CRHPC of Section 3.4.1.1, since these three controllers are almost identical.

In order to compare the robustness of different controllers, the same T-design has been used. The comparative analysis could have been made using the Q-parametrisation scheme introduced in Section 3.5 or just a nominal design ($T = 1$ without the Q parameter). What is relevant in this section are the different robustness bounds provided by the different controllers, whereas the discussion on which is the most convenient robustness enhancing tool is left to the next few sections.

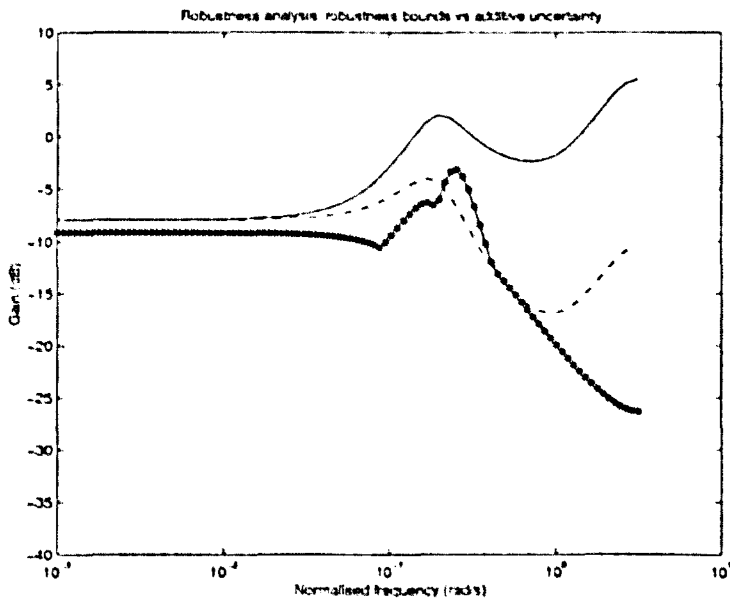


Figure 3.9: Robustness bounds for CRHPC (dashed line) and GPC$^\infty$ (solid line), upper bound of additive system uncertainty for $\Delta_K \in [-0.5, 0.5]$ ("*") and additive system uncertainty for $\Delta_K = 0$ (dotted line)

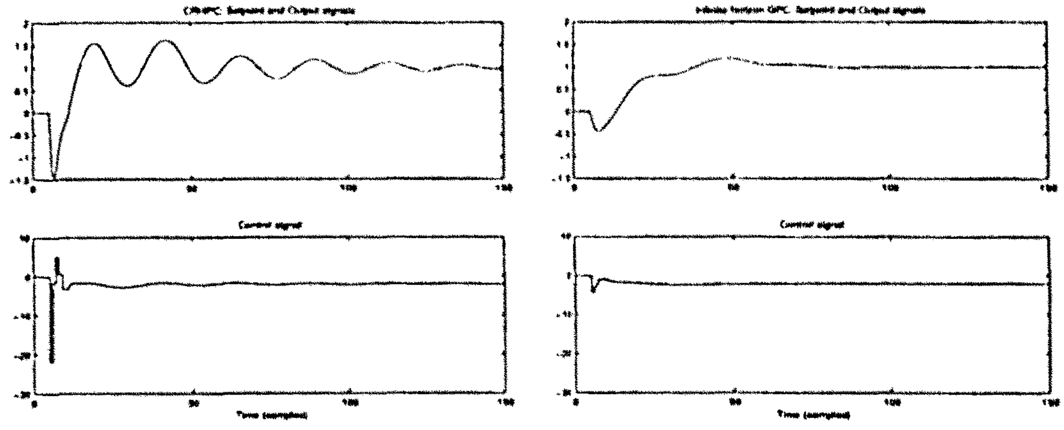
A classical choice of T is given by eqn.3.16, which is known to provide with large robustness bounds for a wide class of systems (Robinson and Clarke, 1991; Soeterboek, 1992; Yoon and Clarke, 1995*a*; Megías, 1996; Megías *et al.*, 1997). Since A is stable, it is included as a factor in T, and the first order polynomial $T^* = (1 - 0.8q^{-1})$ (or

$\nu = 0.8$) has been taken, leading to

$$T(q^{-1}) = A(q^{-1})(1 - 0.8q^{-1}) = 1 - 2.6098q^{-1} + 2.2911q^{-2} - 0.6746q^{-3}. \qquad (3.21)$$

The factor A contributes to T with the roots of $0.9049 \pm 0.1563\jmath$, the modulus of these which is 0.9183, not too near the unit circle. This should provide with a relatively fast disturbance (or modelling error) rejection.

The robustness bounds for the GPC^{∞} and the CRHPC (with $N = 4$) are shown in Fig.3.9. In the best case, $\Delta_K = 0$ (or $K_0 = -0.5$), the bounds provided by both controllers are respected by the modelling errors, and thus the true responses are stable, as displayed in Fig.3.10. Note that the input/output responses obtained with the GPC^{∞} are softer, and the input activity is much lower compared to the CRHPC.

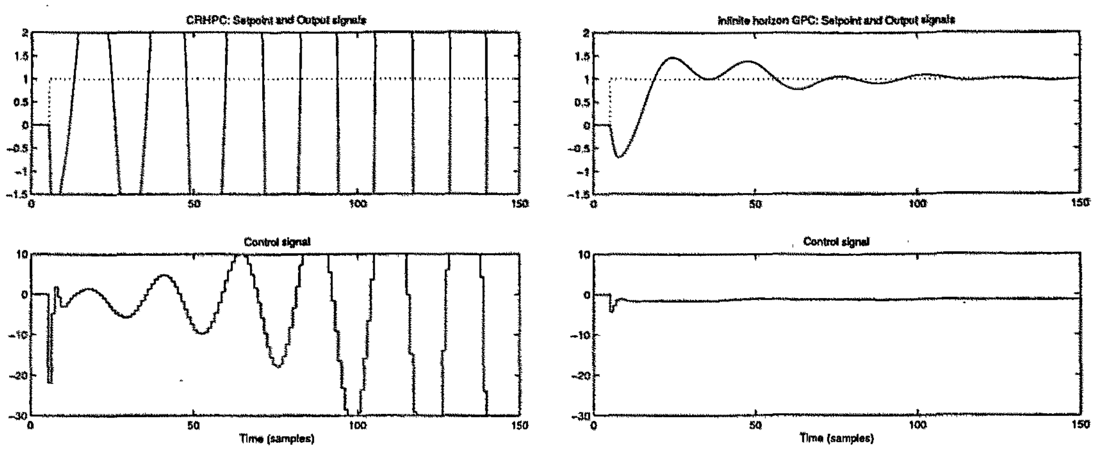


(a) Input/output responses for the CRHPC ($N = 4$) with $\Delta_K = 0$

(b) Input/output responses for the GPC^{∞} with $\Delta_K = 0$

Figure 3.10: True closed-loop behaviour for $\Delta_K = 0$

However, the robustness bound provided by the CRHPC is violated by the upper bound of additive system uncertainty. For example, it is violated for $\Delta_K = 0.5$ (or $K_0 = -0.75$). The consequence of this is shown in Fig.3.11, the responses obtained with the CRHPC are unstable, whereas those provided by the GPC^{∞} are stable (and quite smooth).



(a) Input/output responses for the CRHPC ($N = 4$) with $\Delta_K = 0.5$

(b) Input/output responses for the GPC^{∞} with $\Delta_K = 0.5$

Figure 3.11: True closed-loop behaviour for $\Delta_K = 0.5$

This example illustrates that if system uncertainty is overlooked at the design stage instability may arise. Hence, some (nominal) stabilising laws are preferable to others as they provide smoother nominal (and true) responses and greater robustness bounds. Therefore, the GPC^{∞} (the CRHPC with long prediction/control horizons or the softened CRHPC) appears as a better alternative compared to the CRHPC with short prediction/control horizons as robustness is concerned.

3.4.3 The robustness of 1-norm RHPC formulations

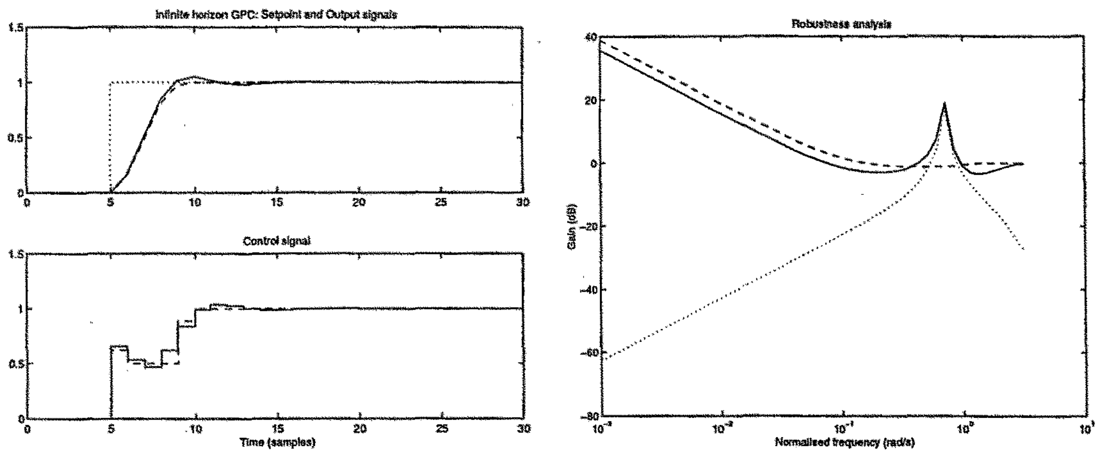
The robustness analysis performed in the previous section is valid only for unconstrained 2-norm RHPC methods, which can be converted into a classical LTI form as shown in Section 3.2.1. However, this is not possible for 1-norm controllers, which are always non-linear. This section illustrates that the robustness-enhancing properties of the polynomial T apply to the 1-norm case too.

Consider the system used in Section 2.4.3 (provided by eqn.A.10) as the nominal plant

$$G(q^{-1}) = \frac{q^{-1}B}{A} = \frac{0.2358q^{-1} + 0.2319q^{-2}}{1 - 1.4835q^{-1} + 0.9512q^{-2}}.$$

Now let the *true* plant be the *unstable system* which arises from multiplying the magnitude of the poles of A by 1.2, and letting the steady-state gain (1) and the zero (-0.9832) be identical to those of G:

$$G_0(q^{-1}) = \frac{q^{-1}B_0}{A_0} = \frac{0.2973q^{-1} + 0.2923q^{-2}}{1 - 1.7802q^{-1} + 1.3698q^{-2}},$$



(a) Nominal input/output responses for the 2-norm GPC$^{\infty}$ (solid) and the GPC$_1^{\infty}$ (dashed): $[N_u, \rho] = [5, 1]$

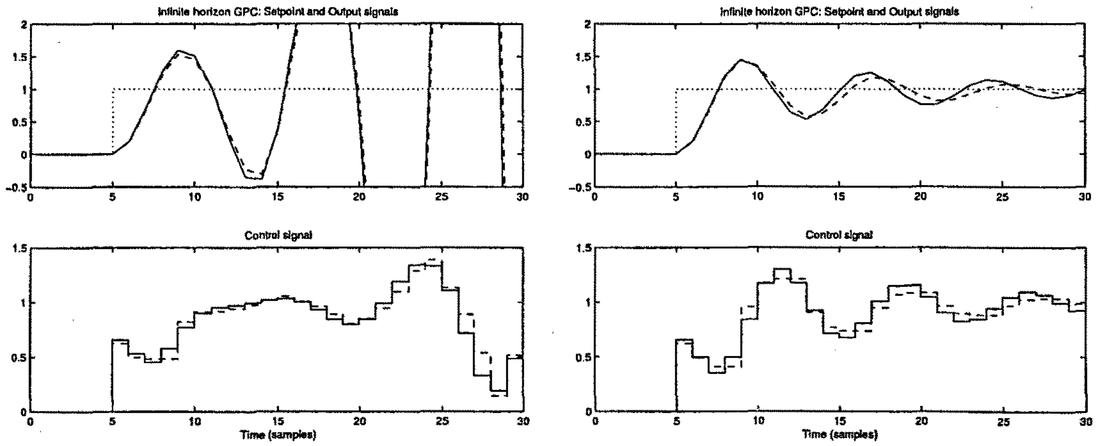
(b) Robustness bounds for T_1 (dashed) and T_2 (solid) and inverse multiplicative uncertainty (dotted)

Figure 3.12: Nominal responses and robustness analysis

The GPC$^{\infty}$ can be designed using the tuning knobs $[N_u, \rho] = [5, 1]$, which provide the nominal input/output responses displayed in Fig.3.12(a) for both 1-norm (which can be implemented simply as the QGPC$_1^{\infty}$ as discussed in Section 2.4.3) and 2-norm formulations. The nominal closed-loop behaviour is almost indistinguishable for both controllers.

To apply the small gain theorem, it must be taken into account that there is a changing number of unstable poles from G (none) to G_0 (two). Hence, an inverse multiplicative uncertainty (which is stable in this case) has been chosen. Two different classical choices have been tried for the observer polynomial: $T = T_1 = A(1 - 0.9q^{-1})$ and $T = T_2 = (1 - 0.9q^{-1})$. In this case the roots of A are $0.7418 \pm 0.6333\jmath$ with modulus 0.9753, *i.e.* very close to the unit circle. Thus, the factor A should be avoided since it is certain to lead to sluggish disturbance rejection.

Fig.3.12(b) shows that the robustness bound obtained with the T_1 (dashed) is violated, whereas that provided by the T_2 (solid) is respected. This example illustrates that including A (when it is stable) as a factor of T is *not always* beneficial, not only as disturbance rejection is concerned, but also as the robustness bound is taken into account.



(a) True input/output responses for the 2-norm GPC^{∞} (solid) and the GPC_1^{∞} (dashed) with $T = T_1$

(b) True input/output responses for the 2-norm GPC^{∞} (solid) and the GPC_1^{∞} (dashed) with $T = T_2$

Figure 3.13: True closed-loop behaviour for the 2-norm GPC^{∞} and the GPC_1^{∞}

The result shown in Fig.3.12(b) guarantees that the 2-norm controller is stable for the true plant when $T = T_2$. However, as shown in Fig.3.13(a), stability is not achieved with $T = T_1$.

For the 1-norm controller, the robustness bounds cannot be obtained in this way, since it cannot be posed in the LTI form. But, as the nominal closed-loop behaviour of the GPC_1^{∞} is almost indistinguishable from the (2-norm) GPC^{∞}, it is expected that T

contributes in an analogous way to robustness for both controllers. This conjecture is confirmed by the results displayed in Fig.3.13, where it is shown that the true responses are very similar for the 1-norm and the 2-norm formulations. Hence, the true closed-loop systems for both the GPC_1^∞ and the (2-norm) GPC^∞ are stable with $T = T_2$ and unstable with $T = T_1$.

3.5 The Q-parametrisation method

Section 3.3 describes a method to increase the robustness bound based on a heuristic choice of the polynomial T, and Section 3.4.2 uses those tuning rules to achieve large robustness margins. However, the heuristic T-design methods are not systematic and do not provide *a priori* guarantees that the associated robustness margin is large enough for a given uncertainty bound, as remarked in (Ansay and Wertz, 1997). As an alternative to this procedure, the Q-parametrisation scheme (often referred to as Youla-Kučera parametrisation also) is introduced in this section.

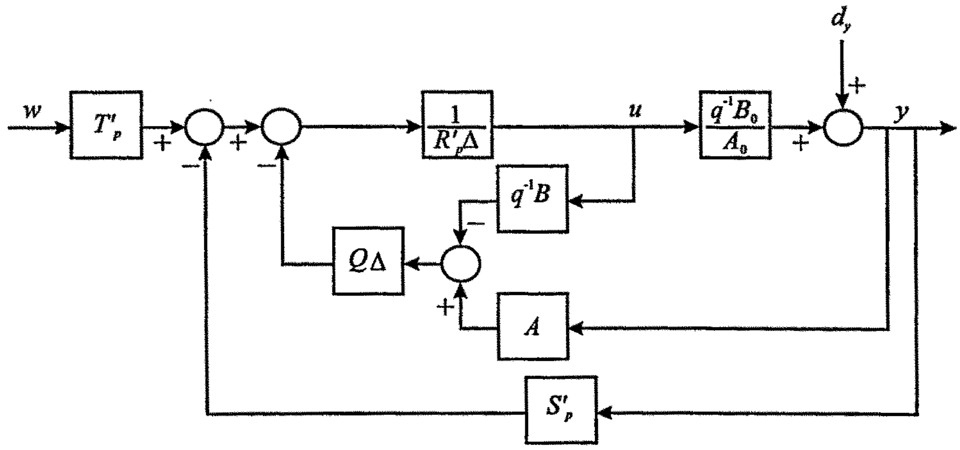


Figure 3.14: Block diagram for the Q-parametrisation scheme

The Q-design methodology, described *e.g.* in (Kouvaritakis *et al.*, 1992; Yoon and Clarke, 1995*a*; Hrissagis *et al.*, 1996; Ansay and Wertz, 1997), is based on the *internal model control* (IMC) scheme of Fig.3.14, where the 2-DOF polynomials R'_p, S'_p and T'_p

are computed taking T to be 1. This choice does not affect the nominal closed-loop transfer functions, since the factor T cancels out in the nominal case, as remarked in Section 3.2.1. The characteristic polynomial TP_c simply reduces to P_c in this case, and can be obtained as

$$P_c = AR'_p\Delta + q^{-1}BS'_p.$$

The block diagram of Fig.3.14 parametrises *all* the controllers which lead to the same nominal closed-loop transfer functions, leaving Q as a free parameter which may be used to enhance robustness. Q is often referred to as the Youla-Kučera parameter. To ensure integral action a factor Δ is explicitly included in Q, which is also bound to be stable.

Remark 3.6 Actually, as discussed in (Yoon and Clarke, 1995*a*), the robustness enhancing method via the polynomial T is nothing but a particular case of this Q-based structure, if Q takes the form

$$Q(q^{-1})\Delta = \frac{M_T(q^{-1})\Delta}{T(q^{-1})},$$

and $M_T(q^{-1})$ is a T-dependent polynomial. □□□

As remarked in (Kouvaritakis *et al.*, 1992; Yoon and Clarke, 1995*a*; Hrissagis *et al.*, 1996), Q can be set, according to the small gain theorem, by minimising the $\mathcal{H}_\infty$ norm of the weighted mixed sensitivity function:

$$\left\|U_y(q^{-1})W_a(q^{-1})\right\|_\infty,$$

where W_a is a weighting transfer function which bounds additive uncertainty and describes the frequency ranges for which the unmodelled dynamics are dominant, and U_y is the control sensitivity at the output, *i.e.* the transfer function between the output disturbance and the control input. For the scheme in Fig.3.14, U_y can be obtained as

$$U_y(q^{-1}) = \frac{A(q^{-1})\left[S'_p(q^{-1}) + A(q^{-1})Q(q^{-1})\Delta\right]}{P_c(q^{-1})}.$$

And the robustness bound is then provided by $|1/U_y|$, as shown in Section 3.2.4.1:

$$|\Delta_a| < \left|\frac{P_c}{A}\right| \left|\frac{1}{S_p' + AQ\Delta}\right|, \ \forall \omega_n \in [0, \pi]. \tag{3.22}$$

If $q^{-1}B/A$ is stable and has no poles on the unit circle, there exists a closed-form analytical solution for the $\mathcal{H}_\infty$ optimisation problem described above, and the optimal Q (Yoon and Clarke, 1995*a*) can be computed according to the next theorem.

Theorem 3.2 (Yoon and Clarke, 1995*a*) *Suppose that the polynomial $A(q^{-1})$ is stable, i.e. $A(q^{-1}) \neq 0$ for $|q| \geq 1$. Then a transfer function Q^{opt} described by*

$$Q^{\text{opt}}\Delta = \frac{-W_a A S_p' + \frac{W_a(1)A(1)S_p'(1)}{P_c(1)} P_c}{W_a A^2}, \tag{3.23}$$

minimises $\|U_y(q^{-1})W_a(q^{-1})\|_\infty$.

Proof: See (Francis, 1987; Yoon and Clarke, 1995*a*) for a proof of this theorem. ∇∇∇

For $Q = Q^{\text{opt}}$ the RS condition for additive uncertainty can be written as

$$|\Delta_a| < |W_a| \left|\frac{P_c(1)}{W_a(1)A(1)S_p'(1)}\right|, \ \forall \omega_n \in [0, \pi], \tag{3.24}$$

that is, the robustness bound follows the shape of W_a shifted by a constant which completely determines the RS margin at low frequencies.

On the other hand, if the nominal system is unstable, any of the procedures described in (Kouvaritakis *et al.*, 1992) or in (Hrissagis *et al.*, 1996) can be applied to compute the optimal Q^{opt}.

A different set of guidelines for tuning the parameter Q can be found in (Ansay and Wertz, 1997), where the choice

$$Q^*\Delta = \frac{S_p'}{A}\frac{M^*\Delta}{C^*}, \tag{3.25}$$

is suggested for stable nominal systems. C^* stands for any stable (Hurwitz) polynomial and M^* can be freely chosen. If A is unstable a different choice of Q^* is suggested in

(Ansay and Wertz, 1997), which is not included here for brevity. The formula for Q^* of eqn.3.25 combined with the robustness bound given in eqn.3.22 leads to the following RS condition for additive uncertainty:

$$|\Delta_a| < \left|\frac{P_c}{A\left(S_p' + AQ^*\Delta\right)}\right| = \left|\frac{P_c}{A\left(S_p' + A\frac{S_p'}{A}\frac{M^*\Delta}{C^*}\right)}\right| = \left|\frac{P_c}{AS_p'}\right|\left|\frac{1}{1+\frac{M^*\Delta}{C^*}}\right|, \quad (3.26)$$

for all $0 \le \omega_n \le \pi$. Now, the filter

$$\frac{1}{1+\frac{M^*\Delta}{C^*}}, \quad (3.27)$$

is totally independent of the term $P_c/(AS_p')$ and can be designed, separately, at a second step. This is a clear advantage with respect to the heuristic T-based methods, since the robustness bound for the T-based methods, provided by

$$|\Delta_a| < \left|\frac{P_c}{AS_p}\right| |T|\,,\ \forall\omega_n \in [0,\pi],$$

does not satisfy this property. The first and the second terms cannot be designed individually, since S_p depends on the polynomial T.

In (Ansay and Wertz, 1997), the simplest possible high-pass filter with unit steady-state gain is suggested for the second term of eqn.3.26, *i.e.*

$$\frac{1}{1+\frac{M^*\Delta}{C^*}} = \frac{1-\nu q^{-1}}{1-\nu},$$

for $0 \le \nu < 1$, which leads to the choice

$$\frac{M^*\Delta}{C^*} = \frac{-\nu\Delta}{1-\nu q^{-1}}, \quad (3.28)$$

and eqn.3.26 becomes the condition

$$|\Delta_a| < \left|\frac{P_c}{AS_p'}\right|\left|\frac{1-\nu q^{-1}}{1-\nu}\right|,\ \forall\omega_n \in [0,\pi].$$

As remarked in (Ansay and Wertz, 1997), a higher order filter can be chosen for eqn.3.27. This possibility is exploited here and in (Megías *et al.*, 1999*a*) in order to provide with a consistent comparison with the T-design method presented in Section 3.3. The filter of eqn.3.27 can be designed such that a polynomial T appears in the numerator of the robustness bound of eqn.3.26 in an analogous way as occurs in eqn.3.14 for the T-design method. This can be achieved by choosing $C^* = T$ and enforcing a unit steady-state gain in eqn.3.27, *i.e.*

$$\frac{1}{1+\frac{M^*\Delta}{C^*}} = \frac{T}{T(1)},$$

or

$$\frac{M^*\Delta}{C^*} = \frac{T(1)-T}{T}. \tag{3.29}$$

For this choice, the RS of eqn.3.26 can be written as

$$|\Delta_a| < \left|\frac{P_c}{AS'_p}\right|\left|\frac{T}{T(1)}\right|, \; \forall\omega_n \in [0,\pi],$$

the second term of which can be designed independently from the first one.

3.5.1 Robust design of GPC$^\infty$ through the parameter Q

In this section the Q-parametrisation is used to enhance the robustness of the nominal GPC$^\infty$ presented in Section 3.4.1.4. The results are compared with the heuristic T-design method used in Section 3.4.2 (with the choice of T specified in eqn.3.21).

For the example introduced in Section 3.4.2, the weighting W_a can be chosen as

$$W_a(q^{-1}) = \frac{1-0.8q^{-1}}{0.2}, \tag{3.30}$$

since:

1. This selection of W_a provides with an upper bound of additive system uncertainty, and the magnitude of this weighting transfer function increases at the frequencies at which uncertainty is greater, as shown in Fig.3.15.

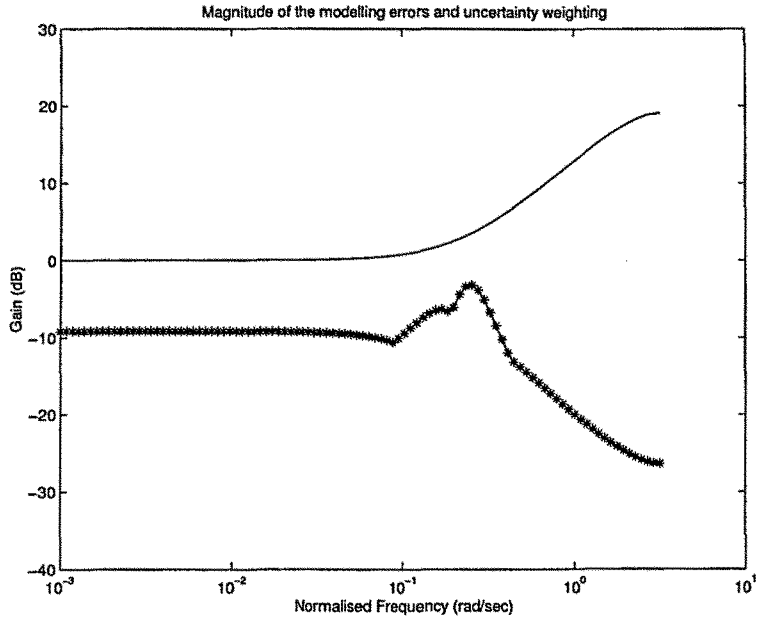


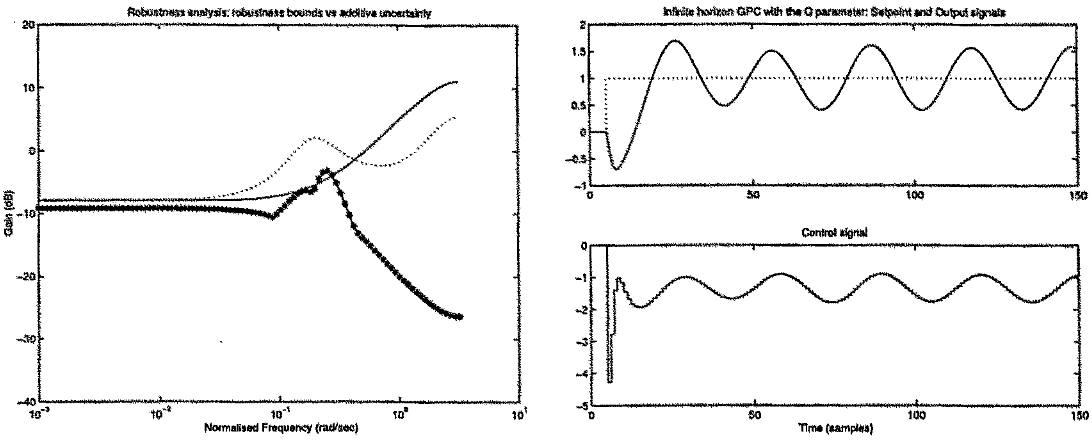
Figure 3.15: Upper bound of additive uncertainty ("*") and uncertainty weighting W_a (solid)

2. This choice of W_a contributes with a factor $(1 - 0.8q^{-1})$ in the numerator of the robustness bound of eqn.3.24. Hence a "fair" comparison with the T-design used in eqn.3.21 is possible, as $(1 - 0.8q^{-1})$ is a factor of T which appears in the numerator of the robustness bound of eqn.3.14. In addition, A cancels out in the right-hand side of both eqn.3.14 wit $T = A(1 - 0.8q^{-1})$ and 3.22 with Q^{opt} as obtained below.

The Q^{opt} computed as per eqn.3.23 for W_a as chosen in eqn.3.30 is given by

$$Q^{\mathrm{opt}} = \frac{10^3\,(0.1773 - 0.7798q^{-1} + 1.3756q^{-2} - 1.2175q^{-3} + 0.5406q^{-4} - 0.0963q^{-5})}{1 - 4.4196q^{-1} + 7.8574q^{-2} - 7.0216q^{-3} + 3.1528q^{-4} - 0.5689q^{-5}}.$$

Fig.3.16(a) compares the robustness bound provided by the Q-parametrisation with Q^{opt} to the heuristic T-design procedure. It is worth pointing out that the former is not only lower than latter within the worst-case uncertainty frequency range, but also violated by the modelling errors. The consequence of this, shown in Fig.3.16(b), is that



(a) Robustness bounds for the Q-parametrisation with Q^{opt} (solid), for the T-design (dotted) and upper bound of additive uncertainty ("*")

(b) True input/output responses for the Q-parametrisation with Q^{opt} and $\Delta_K = 0.5$

Figure 3.16: Robustness bound and closed-loop behaviour for the Q-parametrisation with Q^{opt}

the true input/output responses permanently oscillate for $\Delta_K = 0.5$ in the system of Fig.3.14 with $Q = Q^{\text{opt}}$.

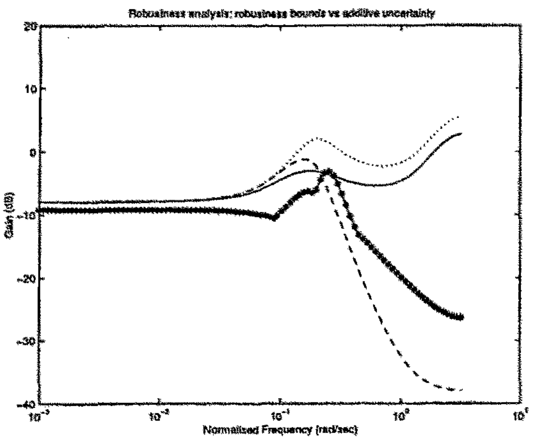
Two alternative choices of Q, suggested by eqn.3.25, have been analysed too. Firstly, Q_1^* has been chosen according to eqn.3.28 with $\nu = 0.8$, in such a way that a factor $(1 - 0.8q^{-1})$ appears in the numerator of the robustness bound:

$$\begin{aligned} Q_1^*\Delta &= \frac{S_p'}{A}\frac{-\nu\Delta}{1-\nu q^{-1}} \\ &= \frac{142.2735 - 395.3817q^{-1} + 367.3608q^{-2} - 114.2526q^{-3}}{1 - 2.6098q^{-1} + 2.2911q^{-2} - 0.6746q^{-3}} \end{aligned}$$

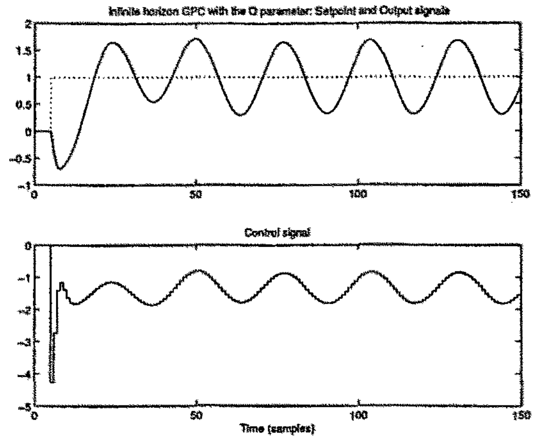
and secondly, Q_2^* has been set using eqn.3.29 (with the value of T specified in eqn.3.21):

$$\begin{aligned} Q_2^*\Delta &= \frac{S_p'}{A}\frac{T(1) - T}{T} \\ &= \frac{10^3\left(0.1767 - 0.7784q^{-1} + 1.3750q^{-2} - 1.2176q^{-3} + 0.5406q^{-4} - 0.0963q^{-5}\right)}{1 - 4.4196q^{-1} + 7.8574q^{-2} - 7.0216q^{-3} + 3.1528q^{-4} - 0.5689q^{-5}}. \end{aligned}$$

Notice that the denominator of Q_2^* coincides exactly with that of Q^{opt}. On the other hand, the difference between the numerators of Q_2^* and Q^{opt} might seem tiny as the *coefficients* are considered, but it is not that small when the *roots* are taken into account.



(a) Robustness bounds for the Q-parametrisation with Q_1^* (dashed), with Q_2^* (solid), for the T-design (dotted) and upper bound of additive uncertainty ("*")



(b) True input/output responses for the Q-parametrisation with Q_2^* and $\Delta_K = 0.5$

Figure 3.17: Robustness bound and closed-loop behaviour for the Q-parametrisation with Q^*

Fig.3.17(a) compares the robustness bounds provided by these two choices of Q^* with the T-design of Section 3.4.2. It is quite remarkable that the robustness bound for Q_1^* is far from required, whereas the one achieved with Q_2^* is always lower than the one provided by the T-design method, and violated by the modelling errors. Fig.3.17(b) shows the worst-case ($\Delta_K = 0.5$) input/output responses attained with[4] Q_2^*, which presents a permanent oscillation similarly as happens for Q^{opt} in Fig.3.16.

This example, which represents the most usual situation as confirmed by the conclusions of (Yoon, 1994; Yoon and Clarke, 1995*a*; Ansay and Wertz, 1997), illustrates that the heuristic T-design method can often provide with better robustness margins compared to the systematic Q-design. Fig.3.18 compares the robustness bounds achieved with T (eqn.3.21), with Q^{opt}, and with Q_2^*. The only one that satisfies the

[4] The closed-loop system for Q_1^* is unstable and its behaviour is not shown in the figure.

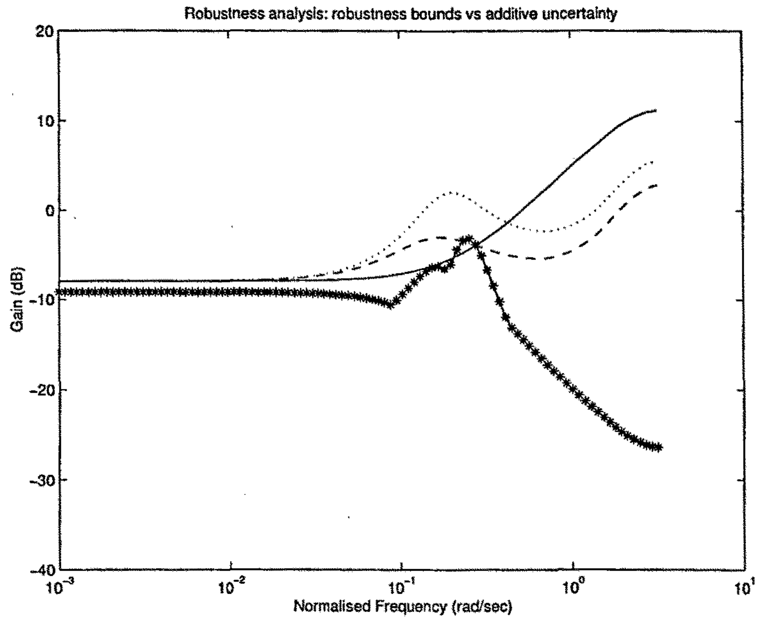


Figure 3.18: Robustness bounds for the Q-parametrisation with Q^{opt} (solid), with Q_2^* (dashed), and for the heuristic T-design (dotted), and upper bound of additive system uncertainty ("*").

small gain theorem condition, and in the end, the only one which provides with a stable closed-loop system for all the plant family, is the heuristic T-design scheme.

However, the T-design is still supported by heuristic rules, which is the main reason for the criticism associated to this procedure. The next section is focused on providing with a systematic framework to obtain a noise polynomial. This procedure is intended to overcome the drawbacks (lack of systematisation) of the T-based schemes, preserving all the advantages (greater robustness margins) pointed out throughout this section.

3.6 The T-optimisation method

The idea introduced in this section is to choose T not heuristically, but by means of an optimisation criterion. Two different criteria are suggested depending on which uncertainty description, additive or inverse multiplicative, is chosen.

If an additive uncertainty is used, the robustness bound provided by the small gain theorem is given by eqn.3.14:

$$|\Delta_a| < \left|\frac{P_c}{A}\right| \left|\frac{T}{S_p}\right|, \; \forall \omega_n \in [0, \pi].$$

Thus, the following problem provides a convenient optimisation criterion to increase the robustness bound:

$$T^{\text{opt}} = \arg \min_{\text{roots}(T)} K_{\text{Robust}} \text{ subject to } \max |\text{roots}(T)| \leq r, \tag{3.31}$$

with

$$K_{\text{Robust}} = \sum_{\omega_k} \left| \frac{A}{P_c} \frac{S_p}{T} W_a \right|^2, \tag{3.32}$$

where roots(T) denotes the set of roots of T, ω_k is a *chosen* set of normalised frequencies in $[0, \pi]$, W_a is a weighting in the frequency domain which bounds additive uncertainty, and $0 < r < 1$ is the maximum allowed modulus for the roots of T. The solution to this problem maximises the right-hand side of eqn.3.14, and hence a greater robustness bound is expected.

Notice that this method is independent from the fact that A is stable or not, which is a clear advantage compared to the heuristic design and to the Q-parametrisation. In addition, the solution depends on three parameters selected by the designer, namely the degree of T, the maximum root radius r and the frequency weight W_a.

For inverse multiplicative uncertainty, the small gain theorem condition is written in eqn.3.15:

$$|\Delta_{im}| < \left|\frac{P_c}{A}\right| \left|\frac{T}{R_p \Delta}\right|, \; \forall \omega_n \in [0, \pi].$$

and thus, the optimisation criterion for T can be posed as eqn.3.31 if K_{Robust} takes the form

$$K_{\text{Robust}} = \sum_{\omega_k} \left| \frac{A}{P_c} \frac{R_p \Delta}{T} W_{im} \right|^2, \tag{3.33}$$

where W_{im} is a weighting in the frequency domain which bounds inverse multiplicative uncertainty.

The choice of the frequency weighting, W_a for additive uncertainty and W_{im} for inverse multiplicative uncertainty, should emphasise the frequencies at which the modelling errors are greater. A bound of uncertainty, analogous to that used in the Q-parametrisation of Section 3.5 is suggested. In fact, the closer the bound is to the maximum modelling errors, the better results arise, since the magnitude of the robustness bound would be increased exactly at the required frequencies.

The objective of the constraint $\max|\text{roots}(T)| \leq r$ enforced in the problem of eqn.3.31 is twofold. Firstly, it ensures that a stable T is obtained. Secondly, it can be used to get rid of possible "slow" roots in T, which can decrease the speed of disturbance rejection, as remarked in Section 3.2.1 and (Robinson and Clarke, 1991). Moreover, as discussed in Section 3.3 and in (Megías, 1996; Megías *et al.*, 1996; Megías *et al.*, 1997), the choice $T = A\Delta$ for stable A leads to $|S_p/T| = 0$, and hence to $K_{\text{Robust}} = 0$ (in eqn.3.32) together with an infinite robustness bound. This possibility is not convenient because the noise model in eqn.2.1 would be cancelled, resulting in an open loop ($S_p = 0$ in the feedback path of Fig.3.1). The constraint in the problem of eqn.3.31 also avoids such a situation.

The sum in eqn.3.32 or eqn.3.33 can be thought of as an approximate integral criterion. In (Megías, 1996; Megías *et al.*, 1997) some criteria which resemble K_{Robust} were presented, *e.g.* the optimisation of $|S_p/T|$, $|AS_p/T|$ or $|WAS_p/T|$. Although the ideas outlined there come from a different approach, the results are similar to the ones presented here. However, those criteria do not include the nominal characteristic polynomial P_c in the optimisation, and thus, are biased and suboptimal.

3.6.1 Robust design of GPC$^\infty$ using the T-optimisation

In this section the optimisation method described above is used to enhance the robustness for the GPC$^\infty$ used in the example provided in Sections 3.4.2 and 3.5.1, and the results are compared with both the heuristic T-design and the Q-parametrisation methods.

The criterion for additive uncertainty of eqn.3.32 has been optimised using the following options:

1. $r = 1 - 10^{-4}$.
2. As an additional constraint, one of the roots of T has been fixed to 0.8, *i.e.* $0.8 \in \text{roots}(T)$ or
$$T^{\text{opt}} = T'(1 - 0.8q^{-1}),$$
to provide a consistent comparison with the T-design in Section 3.4.2 and the Q parameters of Section 3.5.1.
3. Fifty (normalised) frequencies w_k have been chosen in the range $[10^{-2}, \pi]$ rad/s.
4. The *same* frequency weighting W_a used for the Q- parametrisation method, *i.e.*
$$W_a = \frac{1 - 0.8q^{-1}}{0.2},$$
use has been taken.
5. The simplex search optimisation algorithm introduced by Nelder and Mead (1965) has been used for finding the optimal value.

In addition, three different possibilities have been analysed for the factor T':

- T_1^{opt}: $\deg(T') = 2$ with complex-conjugate roots.

- T_2^{opt}: $\deg(T') = 2$ with real roots.
- T_3^{opt}: $\deg(T') = 4$ with two pairs of complex-conjugate roots.

For the first two choices, $\deg(T^{\text{opt}}) = 3$, the same as the heuristically tuned T of eqn.3.21. On the other hand, the latter choice is suggested from the relationship between the polynomial T and the parameter Q, provided in Remark 3.6. If Q is chosen as Q^{opt} or Q_2^* in Section 3.5.1, then the denominator of Q is of the fifth order. This seems to point out to a polynomial T of the fifth degree. However, the robustness bound of eqn.3.22 is such the factor A^2 of the Q denominator (for Q^{opt} and Q_2^*) cancels out. This cancelation takes place in eqn.3.14 when T has A, instead of A^2, as a factor. Despite that, for the sake of completeness, a fourth order polynomial has been tested for T'.

The optimisation performed as described above yields the following polynomials:

$$
\begin{aligned}
T_1^{\text{opt}} &= (1-0.8q^{-1})(1-0.8803q^{-1})^2, \\
T_2^{\text{opt}} &= (1-0.8q^{-1})(1-0.6815q^{-1})(1-0.9553q^{-1}), \\
T_3^{\text{opt}} &= (1-0.8q^{-1})(1-0.8756q^{-1})^2\left(1-[-0.0205 \pm 0.1161\jmath]q^{-1}\right).
\end{aligned}
$$

Notice that the complex-conjugate roots of T_1^{opt} turn out to be a double real root after the optimisation. The value of K_{Robust} obtained with T_2^{opt} is a 7% lower than the one provided by T_1^{opt}, however, one of the roots of T_2^{opt} is located very near the unit circle, which is negative for the speed of disturbance rejection (Clarke and Mohtadi, 1989; Robinson and Clarke, 1991). To prevent such kind of problems, the optimisation could be constrained to roots inside a circle of a given radius, *e.g.* choosing $r = 0.8$.

Finally, it is observed that the modulus of the (second) pair of complex-conjugate roots of T_3^{opt} is tiny compared to the rest. Hence, they contribute with very fast dynamics which can be neglected. The three remaining roots are very approximately the same as those of T_1^{opt}, thus it can be concluded that $T_3^{\text{opt}} \approx T_1^{\text{opt}}$. Needless to say, the values of K_{Robust} are almost identical for T_3^{opt} and T_1^{opt}. This example illustrates that the optimisation procedure leads to equivalent solutions regardless the degree of T, which is a very important property of this method, as the result is not to be affected by the over-parametrisation of T. The unnecessary degrees of freedom given to T are zeroed by the process of optimisation. In addition, this result points out that $\deg(T) = \deg(A) + 1$ seems to be a convenient choice, at least for stable systems when $N_1 = 1$, which is consistent with the results presented in (Soeterboek, 1992; Yoon and Clarke, 1995*a*; Megías *et al.*, 1997).

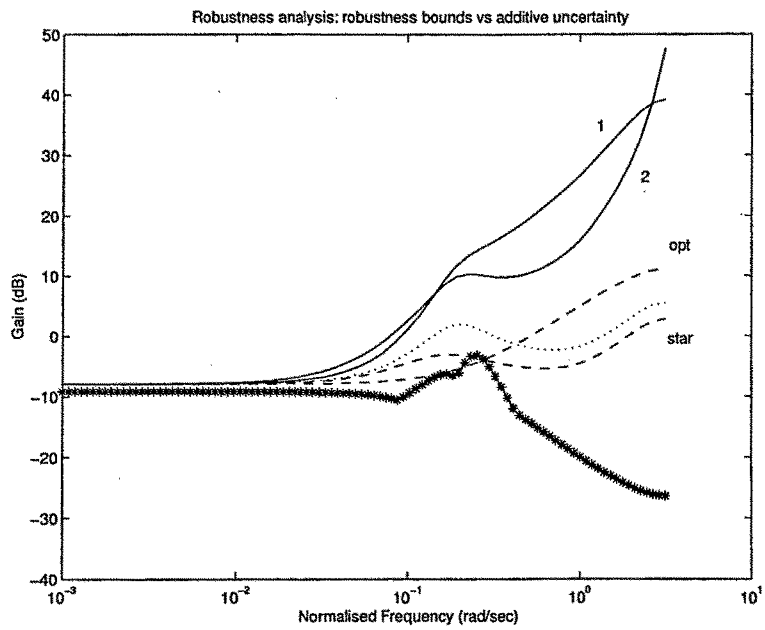
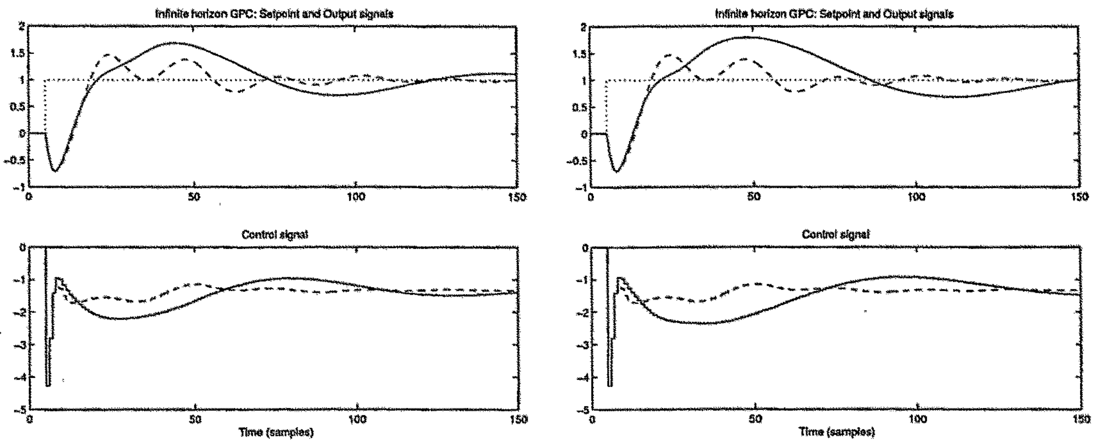


Figure 3.19: Robustness bounds for the T-optimisation (solid lines), T-design (dotted line), Q-parametrisation (dashed lines), and system uncertainty for $\Delta_K = 0.5$ (dash-dotted line).

The robustness bounds obtained with the heuristic T-design of Section 3.4.2, the Q-design presented in Section 3.5.1 for Q^{opt} (labelled "opt") and Q_2^* (labelled "star"), and the T-optimisation method, labelled "1" and "2" for T_1^{opt} and T_2^{opt} respectively, are compared in Fig.3.19. T_3^{opt} is, as already remarked, similar to T_1^{opt}, and thus provides with analogous results which are not shown in Fig.3.19 for clarity. The polynomials T_1^{opt} and T_2^{opt} provide with large robustness bounds, larger than those accomplished with the other methods, ensuring robust stability for the whole family of true systems. The T-optimisation has then been shown to provide with a systematic method to overcome the robustness margins obtained with the heuristic T-design and the (systematic) Q-parametrisation methods.



(a) True input/output responses with T_1^{opt} (solid) and $T = A(1 - 0.8q^{-1})$ (dashed) for $\Delta_K = 0.5$

(b) True input/output responses with T_2^{opt} (solid) and $T = A(1 - 0.8q^{-1})$ (dashed) for $\Delta_K = 0.5$

Figure 3.20: True closed-loop behaviour for the GPC$^\infty$ with $T = A(1 - 0.8q^{-1})$, T_1^{opt} and T_2^{opt}

Fig.3.20 compares the true closed-loop responses obtained with T_1^{opt}, T_2^{opt} and the heuristically tuned $T = A(1-0.8q^{-1})$. The larger robustness bounds obtained with the optimal observer polynomials (Fig.3.19) result in more sluggish responses compared to the heuristic design, as it was expected. What is more, it can be observed that the closed-loop behaviour obtained with T_2^{opt} is even more sluggish than the one provided by T_1^{opt}. The reason for this is that one of the roots of T_2^{opt} (0.9553) is located quite close to the unit circle.

3.7 Concluding remarks

In this chapter the robustness of unconstrained stabilising RHPC methods is analysed. If no inequality constraints are imposed, a classical approach to robustness is possible, since these controllers can be converted into an equivalent 2-DOF LTI form. The classical robust control theory can then be applied, and the small gain theorem is used to derive robust stability conditions in Section 3.2.4 for the most usual uncertainty representations.

The robustness of CRHPC, GPC$^\infty$ and a "softened" CRHPC (a scheme with soft equality constraints) are compared through several experiments in Section 3.4.2. Although these three controllers can produce similar results for particular choices of the tuning knobs, perhaps the GPC$^\infty$ stands out as the best alternative, since it provides with smooth responses and large robustness bounds for small values of N_u, avoiding the numerical problems associated with the CRHPC (Rossiter and Kouvaritakis, 1994). Although the "softened" CRHPC can provide with a similar behaviour, a previous analysis on nominal stability (γ-design) must be carried out.

Additive uncertainty is the most frequently used representation in robust predictive control developments (Robinson and Clarke, 1991; Yoon and Clarke, 1995*a*; Ansay and Wertz, 1997; Megías *et al.*, 1999*a*). However, this description can only be used when the number of unstable poles for all the plant family does not change. This limitation can be overcome using an inverse multiplicative uncertainty, as shown in Section 3.4.3.

It is also evidenced that the robustness-enhancing properties of the polynomial T extend to the 1-norm case. Although the small gain theorem cannot be applied for 1-norm formulations since an equivalent LTI structure of these controllers cannot be found, it has been shown, by means of an example, that the robustness-enhancing properties are identical for 1-norm and 2-norm controllers.

The problem of increasing the bound of the small gain theorem condition when robustness is the priority has been addressed. A new scheme, T-optimisation, has been formulated and compared with two existing methods to enhance robustness, the heuristic T-design and the Q-parametrisation. The T-optimisation is based on the polynomial T but, unlike the T-design, is *not supported by heuristics* but on optimising a quadratic criterion on robustness, thus it is *systematic*. The optimisation criterion is *not* the $\mathcal{H}_\infty$ norm of a sensitivity function, but a quadratic index on the magnitude of a filter. The T-optimisation overcomes the robustness bounds provided by the other procedures even when they optimise a robustness criterion, making it possible to stabilise a wider family of true systems. Thus, this technique can be taken into account when robustness is the priority.

The MIMO case is not considered throughout this chapter because the main aim of this thesis is to achieve robust *constrained* predictive controllers, since constraints are one of the most celebrated advantages of MPC. The unconstrained case presented is this chapter is aimed

1. to show that predictive controllers are quite robust (in a classical sense) although the plant model is a key parameter of these methods, and
2. to point out that some nominally stabilising methods (GPC$^\infty$) are preferable to other (CRHPC) as robustness margins and nominal performance (and also numerical properties) are concerned.