Improving Tracking in Optimal Model Predictive Control

Shukri Salem Dughman

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Sheffield

2018

To Someone special

ABSTRACT

The thesis deals with the improvement in the tracking in model predictive control (MPC). The main motivation is to explore high embedding performance controllers with constraint handling capabilities in a simple fashion. There are several techniques available for effectively using an infinite horizon rather than a finite horizon. First, there has been relatively little discussion so far on how to make effective use of advance information on target changes in the predictive control literature. While earlier work has indicated that the default solutions from conventional algorithms are often poor, very few alternatives have been proposed. This thesis demonstrates the impact of future information about target changes on performance, and proposes a pragmatic method for identifying the amount of future information on the target that can be utilised effectively in infinite horizon algorithms. Numerical illustrations in MATLAB demonstrate that the proposal is both systematic and beneficial.

This thesis introduces several important issues related to model predictive control (MPC) tracking that have been hitherto neglected in the literature, by first deriving a control law for future information about target changes within optimal predictive control (OMPC) for both nominal and constraints cases. This thesis proposes a pragmatic design for scenarios in which the target is unreachable. In order to deal with an unreachable target, the proposed design allows an artificial target into the MPC optimisation problem. Numerical illustrations in MATLAB provide evidence of the efficacy of the proposals.

This thesis extends efficient, robust model predictive control (MPC) approaches for Linear Parameter-Varying (LPV) systems to tracking scenarios. A dual-mode approach is used and future information about target changes is included in the optimisation tracking problem. The controller guarantees recursive feasibility by adding an artificial target as an extra degree of freedom. Convergence to admissible targets is ensured by constructing a robustly invariant set to track any admissible target. The efficacy of the proposed algorithm

is demonstrated by MATLAB simulation.

The thesis considers the tractability of parametric solvers for predictive control based optimisations, when future target information is incorporated. It is shown that the inclusion of future target information can significantly increase the implied parametric dimension to an extent that is undesirable and likely to lead to intractable problems. The thesis then proposes some alternative methods for incorporating the desired target information, while minimising the implied growth in the parametric dimensions, at some possibly small cost to optimality.

Feasibility is an important issue in predictive control, but the influence of many important parameters such as the desired steady-state, the target and the current value of the input is rarely discussed in the literature. At this point, the thesis makes two contributions. First, it gives visibility to the issue that including the core parameters, such as the target and the current input, vastly increases the dimension of the parametric space, with possible consequences on the complexity of any parametric solutions. Secondly, it is shown that a simple re-parametrization of the degrees of freedom to take advantage of allowing steady-state offset can lead to large increases in the feasible volumes, with no increases in the dimension of the required optimisation variables. Simulation with $MATLAB$ 2017a provides the evidence of the efficacy of all proposals.

ACKNOWLEDGMENTS

I thank my God for the grace that I have received on the achievement of the Ph.D. in my field of study.

I would like to express my thanks and appreciation in particular to my thesis supervisor J. Anthony Rossiter. The secret of my success in accomplishing this research is a direct result of his guidance and fruitful views. My communication with him is very useful in educational life. Honestly, I am very happy and consider myself very lucky to work under his supervision.

I would like to thank my wife, my daughter and my sons, who helped me and provided me with success throughout the study period.

I would also like to thank all the members of the department, who contributed to providing me with the opportunity to complete this research

STATEMENT OF ORIGINALITY

Unless otherwise stated in the text, the work described in this thesis was carried out solely by the candidate. None of this work has already been accepted for any degree, nor is it concurrently submitted in candidature for any degree.

Candidate: ______________________

Shukri Salem Dughman

Supervisor: ______________________

Anthony Rossiter

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CARIMA Controlled Auto-Regressive Integrated Moving Average

CLP Closed-Loop Paradigm

d.o.f Degrees Of Freedom

DMC Dynamic Matrix Control

FCCU Fluid Catalytic Cracking Unit

FF Feed-Forward

GMV Generalised Minimum Variance

GPC Generalised Predictive Control

LMI Linear Matrix Inequality

LPV Linear Parameter Varying

LQ Linear Regulator

MAC Model Algorithm Control

MAS Maximum Admissible Set

MCAS Maximum Controlled Admissible Set

MIMO Multi-Input Multi-Output

MPC Model Predictive Control

mp-QP Multi-parametric Quadratic Programming

OMPC Optimal Model Predictive Control

PFC Predictive Function Control

PID Proportional Integral Derivative

PP Pole Placement

QP Quadratic Programming

RHC Receding Horizon Control

RMCAS Robust Maximum Controlled Admissible Set

SISO Single Input Single Output

Chapter 1

INTRODUCTION

In the past, the control was performed manually where operators control certain variables, such as temperature, speed, and other variables available in the industrial field. The challenge with manual control is the need for better quality. With the rise of the industrial revolution, the automatic control system has been built up to enforce the control scheme.

Automatic control theory has played a key role in the development of the industry. This is because of the increase in production processes and quality improvement of manufactured goods.

This control system comprises the main element (controller) which executes the control task. Perhaps one of the most important controller types within the industry is the PID controller [102, 68, 81]. For PID tuning, there are many methods such as Tyreus–Luyben settings, Ziegler–Nichols, and reaction curve methods [155]. However, this controller suffers from some restrictions.

The PID controller is perfect for some scenarios and often fails to control time delayed and multi-variable processes effectively. Also, this controller is unable to handle system constraints systematically. Therefore, there is a demand for control strategies that are better than a PID controller.

Several methods were introduced in 1970's, such as Minimum variance (MV), Smith Predictor control (SPC), Generalised Minimum Variance (GMV), and Pole Placement (PP). Although these methods provide improved performance compared to PID, they suffer from some limitations in certain scenarios [61]. This led to the development of more sophisticated control approaches, and now MPC is one of the most successful advanced control approaches in the industrial field [129, 101, 120, 49]. This is largely due to its ability to handle input and state constraints and multi-variable processes. Furthermore, it has the potential to include feed-forward information systematically rather than as a separate design. The evidence is presented in [120], [13], [127], [29], and [175]. Since then MPC has become a popular control approach in the industrial field. This approach uses a model of the system to produce a prediction of its future behaviour.

The basic concepts of model predictive control are well illustrated and understood, e.g. [145, 167, 19]. A brief illustration of MPC is demonstrated in the following sections.

1.1 MPC strategy

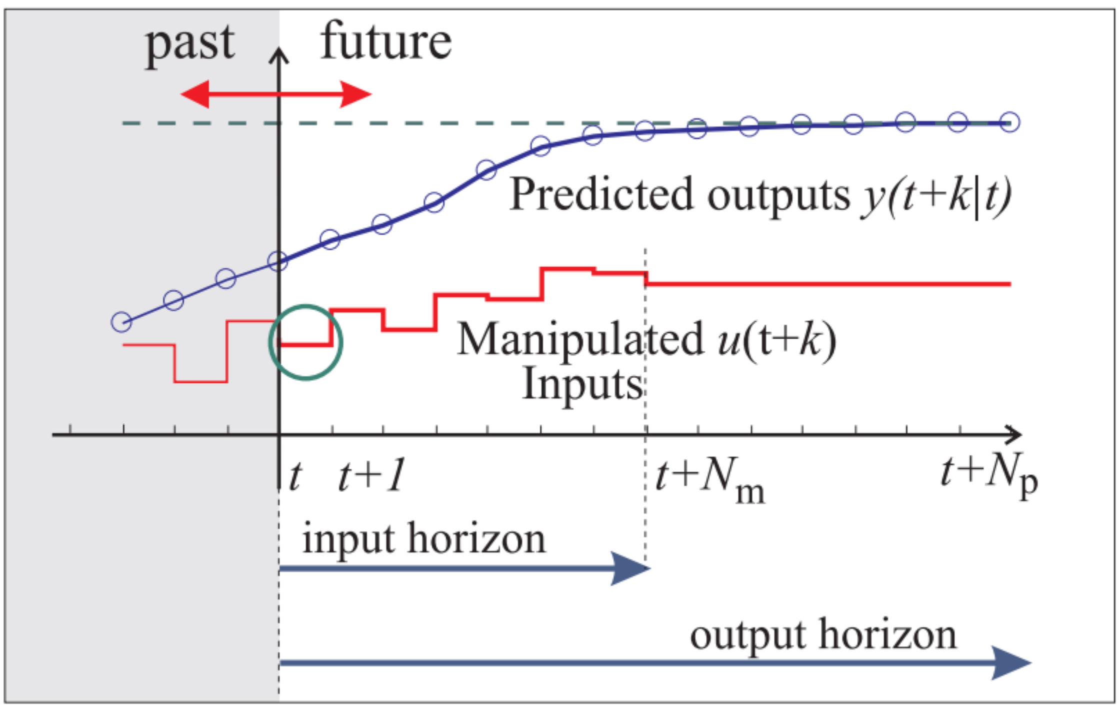

Figure 1.1: MPC strategy[13]

A typical MPC strategy is illustrated in Fig 1.1 [13]. It proposes a sequence of candidate future input moves which are expected to give the best predicted performance, where performance is assessed using a defined performance index. Usually, MPC utilises only the first move of the control candidate sequence, while ongoing measurement and optimisation are used continually to improve the planning for each sample. This philosophy is called receding horizon control (RHC).

1.2 MPC structure

A typical MPC implementation structure is shown in Figure 1.2 [13]. The MPC is based on an internal model which is used to predict the future process outputs. This controller manipulates the difference between the set point (SP) and the feedback process variable (PV) to provide manipulated variables (MV) to the controlled object. The disturbance is rejected by disturbance cancellation inside the controller.

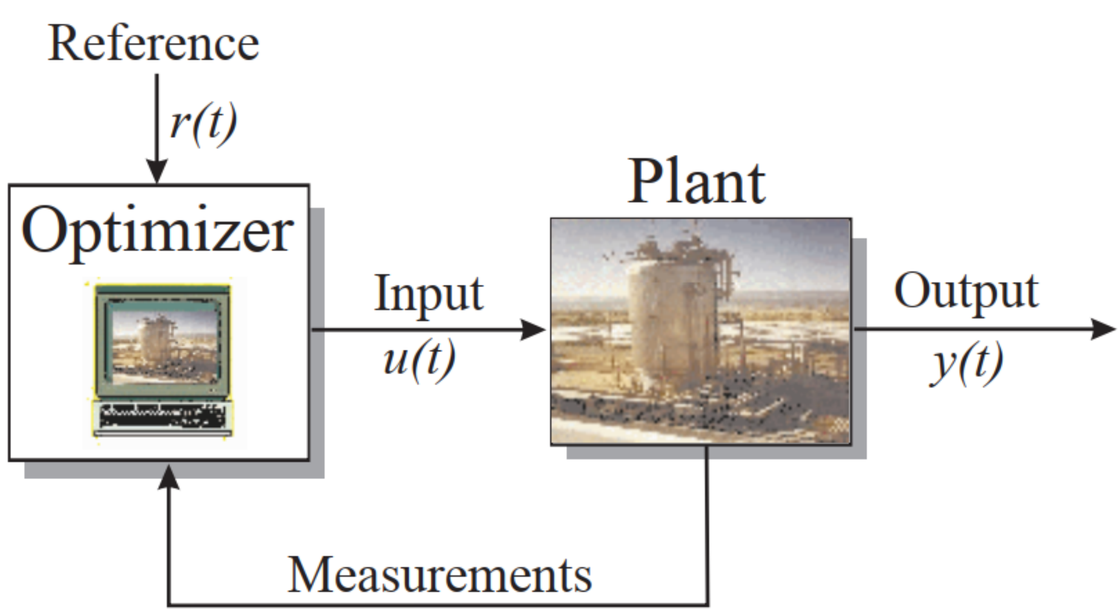

Figure 1.2: Basic MPC structure [13]

1.3 MPC methods

MPC has developed gradually starting from a simple controller such as a minimum variance (MV) introduced in the 1970s to a Generalized Predictive Control (GPC) [27, 26] in the 1980s then to an optimal control in the 1990s, e.g. Model Algorithmic Control (MAC), Dynamic Matrix Control (DMC) [99, 147, 103], Predictive Functional control (PFC) [130, 132, 129], Extended Prediction Self-Adaptive Control(EPSAC) [136], Generalized Predictive Control (GPC) [26, 27, 55], and Internal model control (IMC) [44, 47, 38].

Generalized Predictive Control (GPC) and Dynamic Matrix Control (DMC) were popular MPC algorithms. These MPC strategies were based on modelling the controlled process and prediction over a finite horizon which is chosen with weighting factors as the tuning parameters. The weakness of these algorithms is that the performance can be poor if the tuning parameters are poorly chosen. It is also shown that it is difficult to guarantee asymptotic stability with finite horizons in a predictive control approach. Recently, infinite horizon algorithms have been developed [95], such as Optimal Model Predictive Control (OMPC) or dual-mode algorithms, that tackled some of the weaknesses associated with the finite horizon methods. Since then they have become popular algorithms in the literature. The methods introduced in this section are discussed in [61].

1.4 Research challenges

One key advantage of MPC is it can make systematic use of advance information. However, although in principle MPC can use future information about set point and disturbances systematically, often the default use of this information is poor. An open question in the literature is how this information is used effectively.

Most of the literature on model predictive control (MPC) ignores the use of advance knowledge, and therefore implicitly ignores the feed-forward design in MPC. In a typical dual-mode approach, it is assumed that the target is fixed. However, there are scenarios where it is necessary to consider the time-varying target in the case of using advance knowledge.

Therefore the performance index requires some minor changes to allow for time-varying future targets.

Typical MPC performance indices penalise the errors between the future set point and the predicted output. This is known as a tracking scenario; however, one of the challenges within a tracking scenario is the need to ensure feasibility, in order to guarantee that the class of predictions available to the MPC algorithm can indeed satisfy all of the constraints simultaneously. However, even putting a side issues linked to model uncertainty, feasibility can easily be lost during rapid or large set point changes and disturbance changes, both of which have a strong impact on the terminal constraints [145]. Consequently, there is a strong link between set point tracking and feasibility; the feasibility of the controller may be lost, and the controller becomes ill-defined, or not defined at all [126], in the case of any set point changes. A convenient and essential component for guaranteeing the stability of the MPC algorithms is to ensure feasibility; that is, to ensure the existence of a set of future controls which are within the input constraints and meet suitable terminal constraint. Assuming a suitable underlying MPC approach such as dual-mode [145, 153], a feasibility guarantee is often sufficient to enable a simple guarantee of nominal (and at times robust) closed-loop stability for the controller.

Several authors have tackled this tracking problem, that is the loss of feasibility, by developing modified formulations for the MPC algorithm to deal with different scenarios. Recently, novel cost function formulations have been devised to guarantee feasibility and stability in MPC tracking. This novel cost formulation penalises the deviation from an artificial steady-state target and the terminal state in addition to penalising the deviation of the artificial steady-state target from the true steady-state target, e.g, [143, 137, 86]. They achieved good results ensuring feasibility retention and hence stability guarantees. Nevertheless, the range of solutions and approaches in the literature remains relatively limited.

There is a strong link between the feed-forward and degrees of freedom (d.o.f) used for constraint handling, as these impact on the closed-loop. It has been shown in [144] that it is possible and probably advantageous to use a feed-forward design to deal with constraints, thus shifting the major computational load to an off-line problem. The optimisation of the d.o.f then need only focus on feedback aspects that arise from parameter uncertainty and disturbances. The most appropriate methods for this issue are multi-parametric MPC methods.

1.5 Thesis objectives

There are several objectives related to tracking MPC improvement presented in this thesis. These objectives are discussed briefly in the following subsections.

1.5.1 Fixed feed-forward designs

It has been shown in [145] that the default feed-forward arising from a conventional MPC algorithm may be ineffective because the assumptions implicit in the optimisation are relatively limited and only valid for fixed set points. If the future set point is changing, then the optimisation and degrees of freedom within it need essential modification. It is logical to consider whether a two stage design will provide better choices for the feed-forward; in other words: (i) first design the feedback loop for robust performance and (ii) secondly, design a feed-forward to give optimum tracking, assuming the inner loop is known. Regarding this concept, it is necessary to clarify exactly how the design can be performed and to what extent such a design can handle constraints. Consequently, a key objective is to define an algorithm that can make effective use of a predefined feed-forward; that is, to embed this into the on-line optimisation and evaluate the approach compared to more conventional methods. Another issue to consider is the extent to which the future values of the target can be treated as states in a parametric optimisation. A main objective of this thesis is to improve MPC tracking through the use of dual-mode algorithms that have the potential to be implemented in industrial contexts.

1.5.2 Artificial targets in MPC

In practice, there is a scenario where the constraints are active in the steady-state and this can make the desired target unreachable. Recent work has proposed an artificial target which is reachable, but the challenge here is how to compute and choose this target and incorporate it in the MPC optimization. Therefore, it is logical to devise a novel algorithm that computes and chooses an artificial target and incorporates it into the performance index using both finite and infinite horizon algorithms to show the impact of the terminal constraints on feasibility. Another important and linked issue is to optimise the artificial target using parametric methods.

1.5.3 The robust case

It has been shown that model parameter uncertainty can cause a loss of feasibility, in which case MPC would become undefined; therefore, it is useful to analyse the existing approaches that are robust to parameter uncertainty and then to propose some modifications to tracking approaches, which guarantee this robustness. Recent work has been done on this issue such as a robust invariant set approach but this is limited to specific scenarios. The objective here is to use an artificial target which is more flexible and easier than the conventional approach of tubes [80, 123].

1.6 Supporting publications

The work in this thesis is supported by the following publications :

1.6.1 Conference papers

1. S. Dughman and J. A. Rossiter. A survey of guaranteeing feasibility and stability in MPC during target changes. IFAC-Papers OnLine, 48(8):813818, 2015.

2. S. Dughman and J. Rossiter. Systematic and simple guidance for feed-forward design in model predictive control. 18th International Conference on Control Science and System Engineering, 2016.

3. S. Dughman and J. A. Rossiter. Efficient feed forward design within MPC. In Control Conference (ECC), 2016 European, pages 1341–1346. IEEE, 2016.

4. SS Dughman and JA Rossiter. Efficient Robust Feed-Forward Model Predictive Control with Tracking. UKACC ,11th International Conference, 2016.

5. S. Dughman and J. Rossiter. The feasibility of parametric approaches to predictive control when using far future feed-forward information. In Control and Automation (ICCA), 2017 13th IEEE International Conference on, pages 1101–1106. IEEE, 2017.

6. S. Dughman and J. Rossiter. The impact of the input parameterisation on the feasibility of MPC and its parametric solution. In Proceedings of the European Control Conference. IEEE, 2018.

1.6.2 Journal papers

1. S. Dughman and J. Rossiter. Systematic and effective embedding of feed-forward of target information into MPC. International Journal of Control, pages 1-15, 2017.

1.7 Thesis overview

This section gives an overview of this thesis in the following subsections: The aim of this thesis is to implement algorithms that design an efficient feed-forward for tracking MPC through the use of future information about target and disturbance changes and guaranteeing the feasibility hence ensuring closed-loop stability. Moreover, efficient feed-forward is also extended to be designed for the robust case. Finally, the feed-forward is improved by implementing a multi-parametric quadratic programming method.

Chapter 2, introduces the background on MPC and **Chapter 3**, describes the literature review on MPC algorithms.

Chapter 4, presents the feed-forward designs with the use of future information about target and disturbance changes for dual-mode or Optimal Model Predictive Control (OMPC) algorithms. Efficient algorithms are demonstrated for both unconstrained and constrained cases.

Chapter 5, discusses the artificial targets in MPC demonstrating the use of advance knowledge when the target is unreachable during transient and steady-states.

Chapter 6, presents robust tracking with advance knowledge by proposed an algorithm for Linear Parametric Varying (LPV) systems with the presence of uncertainty.

Chapter 7, demonstrates complexity anf feasibility of a multi-parametric quadratic programming (mp-QP) technique within MPC tracking using advance information of target changes.

Chapter 8, studies the implementation of the proposed OMP feed-forward algorithms to different industrial processes, for both nominal and robust design.

Finally, the conclusions and future work recommendations are presented in **Chapter 9**.

Appendix A, presents some details for **Chapter 7**.

Appendix B, presents state space parameters for large models, presented in **Chapters 4 and 8**.

Chapter 2

BACKGROUND ON MPC

This chapter presents the background on MPC and is divided in two parts. **Part I** introduces some of the components which are essential for MPC implementation. These are presented mathematically in the subsequent sections based on the full state space formulation. Section 2.1 presents the system model description. Predictions based on the use of one step ahead model are detailed in Section 2.2. Common cost functions are presented in Section 2.3. Section 2.4 defines the constraints for finite horizon approaches, while Section 2.5 describes the constraints for an infinite horizon. Section 2.6 presents the optimisation and degrees of freedom.

Part II introduces the stability issue with infinite horizon predictions, together with a background on the dual-mode MPC approach. Section 2.7 discusses the dual-mode MPC approach while, Section 2.8 explores the closed-loop paradigm. The feed-forward compensator is presented in Section 2.9 and multi-parametric quadratic programming is described in Section 2.10. Finally, a summary of this chapter is provided in Section 2.11.

2.1 System Models

This section describes the system modelling assumptions and the different types of predictions.

2.1.1 Typical models

There are two main models used in model predictive control to describe the processes. These are a transfer function model and a state space model. For convenience, the preferred model type used in the MPC is the state space model. This is due to its ability to handle multi-variable systems more easily than a transfer function model. The state space model is assumed to be used for an open-loop stable and/or unstable plants and can be represented as follows.

$$x_{k+1} = Ax_k + Bu_k, \qquad y_k = Cx_k + Du_k \tag{2.1}$$

where x_k, y_k, u_k are states, process output and process inputs respectively and A, B, C, D are the matrices defining the model; in this thesis, we assume a strictly proper system ($D = 0$). An output disturbance d_k can be incorporated into the model with a small modification as:

$$x_{k+1} = Ax_k + Bu_k, \qquad y_k = Cx_k + d_k \tag{2.2}$$

where the disturbance d_k is assumed to be varying slowly and can be estimated.

In this case, the observer models are augmented to include the disturbance in the system dynamics and equation (2.2) can be replaced by:

$$\underbrace{\begin{bmatrix} x_{k+1} \\ d_{k+1} \end{bmatrix}}_{\mathbf{x}_{k+1}} = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} \underbrace{\begin{bmatrix} x_k \\ d_k \end{bmatrix}}_{\mathbf{x}_k} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k; \quad y_k = \begin{bmatrix} C & I \end{bmatrix} \begin{bmatrix} x_k \\ d_k \end{bmatrix} \tag{2.3}$$

The system model can also be described in terms of the control increments as the augmented state space model which is given by:

$$\underbrace{\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} A & B \\ 0 & I \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}}_{\tilde{x}} + \underbrace{\begin{bmatrix} B \\ I \end{bmatrix}}_{\tilde{B}} \triangle u_k \tag{2.4}$$

$$y_k = \underbrace{\begin{bmatrix} C & D \end{bmatrix}}_{\tilde{C}} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} \tag{2.5}$$

To ensure an absence of bias in steady-state predictions, this model should satisfy the prediction consistency condition; that is:

$$y = r \quad \Longrightarrow \quad \triangle u = 0 \tag{2.6}$$

For convenience this thesis uses $\tilde{x} = x$ to reduce the notation.

2.1.2 Steady state model

The model can also be formulated in terms of steady-state input and state as shown below.

Considering the state space model with output disturbance (2.2), the expected steady-state in the future can be defined as follows.

$$\begin{aligned} x_{ss} &= Ax_{ss} + Bu_{ss} \\ y_{ss} &= Cx_{ss} + d_k \end{aligned} \tag{2.7}$$

where x_{ss}, u_{ss}, y_{ss} are the state, input, and output steady-state values, respectively.

The steady-states values x_{ss}, u_{ss} can be computed by solving equation (2.7) as follows.

$$\begin{bmatrix} y_{ss} - d \\ 0 \end{bmatrix} = \begin{bmatrix} C & 0 \\ A - I & B \end{bmatrix} \begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} \tag{2.8}$$

$$\begin{bmatrix} C & 0 \\ A - I & B \end{bmatrix}^{-1} \begin{bmatrix} y_{ss} - d \\ 0 \end{bmatrix} = \begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} \tag{2.9}$$

Hence, one can define the deviation variables as:

$$\hat{x} = x_k - x_{ss}; \ \hat{u}_k = u_k - u_{ss}; \ \hat{y} = y_k - y_{ss} \tag{2.10}$$

where $\hat{x}_k, \hat{u}_k, \hat{y}_k$ are state, input, and output deviation variables respectively.

Using superposition between equation (2.7) and equation (2.10), one can obtain a state space model in terms of the deviation variables as:

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + B\hat{u}_k \\ \hat{y}_k &= C\hat{x}_k \end{aligned} \tag{2.11}$$

It is noted in equation (2.11) that the disturbance d_k has been absorbed in the estimation of the correct steady-state.

Remark 2.1 *To ensure a lack of bias in steady-state prediction, this mode should satisfy that: as the disturbance (d_k) varies, the implied steady-state values of u and x in the model (2.4, 2.5) can move to ensure that (2.6) holds.*

2.2 Predictions

Predictions can be expressed in open-loop or closed-loop fashion. This section presents the different types of predictions, as shown in the following subsections:

2.2.1 Open-loop predictions

For the open-loop fashion, one can consider the common state space model which gives one step ahead predictions:

$$x_{k+1} = Ax_k + Bu_k, \qquad y_{k+1} = Cx_{k+1} + d_{k+1} \tag{2.12}$$

It is assumed that ($d_{k+i} = d_k, \quad i = 0, \ldots, n$) because the disturbance varies slowly and the future disturbance is unknown.

The one-step ahead prediction can be used recursively to find an n-step ahead prediction, as follows.

At the time step k+2, equation (2.12) is given by:

$$\mathbf{x}_{k+2} = Ax_{k+1} + Bu_{k+1}; \qquad y_{k+2} = Cx_{k+2} + d_{k+2} \tag{2.13}$$

Substituting for x_{k+1} into (2.13) gives:

$$\mathbf{x}_{k+2} = A^2x_k + ABu_k + Bu_{k+1}; \qquad \mathbf{y}_{k+2} = Cx_{k+2} + d_{k+2} \tag{2.14}$$

At the time step $k+3$, the prediction using equation (2.14) is given by:

$$x_{k+3} = A^2x_{k+1} + ABu_{k+1} + Bu_{k+2}; \qquad y_{k+3} = Cx_{k+3} + d_{k+3} \tag{2.15}$$

Substituting for x_{k+1} in equation (2.15) gives:

$$x_{k+3} = A^3x_k + A^2Bu_k + ABu_{k+1} + Bu_{k+2};\; y_{k+3} = Cx_{k+3} + d_{k+3} \tag{2.16}$$

A general expression of the n-step ahead predictions can be given as:

$$\begin{gathered} x_{k+n} = A^nx_k + A^{n-1}Bu_k + A^{n-2}Bu_{k+1} + \ldots.. + Bu_{k+n-1} \\ y_{k+n} = Cx_{k+n} + d_{k+n} \end{gathered} \tag{2.17}$$

Assuming $d_{k+i} = d_k, \quad i = 0, \cdots, n$, the state and output predictions can be expressed for a future horizon up to n_y using arrow notation and vectors as:

$$\underbrace{\begin{bmatrix} x_{k+1} \\ x_{k+2} \\ x_{k+3} \\ \vdots \\ x_{k+n_y} \end{bmatrix}}_{\underset{\rightarrow k}{x}} = \underbrace{\begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^n \end{bmatrix}}_{P_x} x_k + \underbrace{\begin{bmatrix} B & 0 & 0 & .. \\ AB & B & 0 & .. \\ A^2B & AB & B & .. \\ \vdots & \vdots & \vdots & \vdots \\ A^{n_y-1}B & A^{n_y-2}B & A^{n_y-3}B & : \end{bmatrix}}_{H_x} \underbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+n_y-1} \end{bmatrix}}_{\underset{\rightarrow k-1}{u}} \tag{2.18}$$

$$\underbrace{\begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ \vdots \\ y_{k+n_y} \end{bmatrix}}_{\underset{\rightarrow k}{y}} = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^n \end{bmatrix}}_{P_y} x_k + \underbrace{\begin{bmatrix} CB & 0 & .. \\ CAB & CB & .. \\ CA^2B & CAB & .. \\ \vdots & \vdots & \vdots \\ CA^{n_y-1}B & CA^{n_y-2}B & : \end{bmatrix}}_{H_y} \underbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+n_y-1} \end{bmatrix}}_{\underset{\rightarrow k-1}{u}} + \underbrace{\begin{bmatrix} d_k \\ d_k \\ d_k \\ \vdots \\ d_k \end{bmatrix}}_{Ld_k} \tag{2.19}$$

Writing equations (2.18) and (2.19) in a compact form, we have:

$$\begin{aligned} \underset{\rightarrow k}{x} &= P_x x_k + H_x \underset{\rightarrow k-1}{u} \\ \underset{\rightarrow k}{y} &= P_y x_k + H_y \underset{\rightarrow k-1}{u} + L d_k \end{aligned} \tag{2.20}$$

where $L = \begin{bmatrix} I \\ I \\ \vdots \end{bmatrix}$, x_k is an augmented state and d_k is the disturbance estimate.

It is shown from equation (2.20) that the prediction structure includes parts (based on the current measurements, x_k) and parts (based on the decision variable, u_k yet to be determined).

2.2.2 Unbiased predictions

Unbiased prediction means consistency between the future output predictions and the system steady-state values. This can be ensured by using different methods, but there is a suitable mechanism used with the state space model that is the steady-state estimates. The idea is to estimate the expected steady-state values for the state, input and disturbance to meet a given steady-state output and then use deviations about this point.

Considering the state space model (2.11), and following equation (2.20), one can define the unbiased state and output predictions in terms of deviation variables as:

$$\begin{array}{r}\underset{\rightarrow k}{\hat{x}} = P_x \hat{x}_k + H_x \underset{\rightarrow k-1}{\hat{u}} \\ \underset{\rightarrow k}{y} = P_y \hat{x}_k + L y_{ss} + H_y \underset{\rightarrow k-1}{\hat{u}}\end{array} \quad ; \quad \underset{\rightarrow k}{y} = \underset{\rightarrow k}{\hat{y}} + L\ y_{ss} \tag{2.21}$$

These predictions of (2.21) are based on the state and future input deviations ($\hat{x}_k$, $\underset{\rightarrow k-1}{\hat{u}}$) and the steady-state output (y_{ss}).

2.2.3 Closed loop prediction

It was demonstrated in Subsection 2.2.1 how to define predictions in an open-loop fashion. This subsection describes the closed-loop predictions for state space models, which can be deployed in MPC algorithms.

Using the typical stabilising state feedback $u_k = -Kx_k$, the equations within the closed-loop during prediction can be given by:

$$x_{k+1} = Ax_k + Bu_k; \quad u_k = -Kx_k. \tag{2.22}$$

Removing the dependent variable u_k gives:

$$x_{k+1} = [A - BK]x_k; \qquad u_k = -Kx_k \tag{2.23}$$

Simulating these forward in time with $\Phi = A - BK$, one gets:

$$\underset{\rightarrow k}{x} = \underbrace{\begin{bmatrix} \Phi \\ \Phi^2 \\ \Phi^3 \\ \vdots \\ \Phi^n \end{bmatrix}}_{P_{cl}} x_k. \tag{2.24}$$

Equation (2.24) can be expressed in compact form as:

$$\underset{\rightarrow k}{x} = P_{cl} x_k \tag{2.25}$$

The corresponding input predictions can be written as:

$$\underset{\rightarrow k}{u} = \underbrace{\begin{bmatrix} -K \\ -K\Phi \\ -K\Phi^2 \\ \vdots \end{bmatrix}}_{P_{clu}} x_k. \tag{2.26}$$

or in a compact form as:

$$\underset{\rightarrow k}{u} = P_{clu} x_k \tag{2.27}$$

The state can be defined after n_c steps to give the form:

$$x_{k+n_c} = P_{cl2} x_k \tag{2.28}$$

where x_{k+n_c} is the state value predicted at $k + n_c$

It is shown in equation (2.28) that the prediction is a function of the current state x_k.

2.2.4 Basic results:

It is common to use state space models in MPC. This section has shown that:

1. It is common to use discrete state space models for predictions.
2. Predictions are based on current measurements and the degree of freedom (d.o.f) .
3. Unbiased predictions can be ensured if the steady-state values (x_{ss}, u_{ss}, y_{ss}) are consistent.
4. Open loop predictions are given by:

$$\underset{\rightarrow k}{\hat{x}} = P_x \hat{x}_k + H_x \underset{\rightarrow k-1}{\hat{u}}$$
$$\underset{\rightarrow k}{y} = P_y \hat{x}_k + L y_{ss} + H_y \underset{\rightarrow k-1}{\hat{u}}$$

5. Closed loop predictions are given by:

$$\underset{\rightarrow k}{x} = P_{cl} x_k, \qquad \underset{\rightarrow k}{u} = P_{clu} x_k$$

2.3 The performance index (cost function)

For a typical MPC algorithm, the control law is based on the optimization of the predicted performance based on a performance index. In this thesis, we use the terminology 'performance index' and 'cost function' interchangeably.

This section will present the common cost functions used in MPC approaches.

2.3.1 Typical cost functions

A typical common performance index, e.g. [145, 143], penalizes the weighted squares of both predicted tracking errors and the control increments/deviations; that is:

$$J = \sum_{i=1}^{n_y} \left\| r_{k+i} - y_{k+i} \right\|_2^2 + \sum_{i=0}^{n_{u-1}} \left\| W(u_k - u_{ss}) \right\|_2^2 + \left\| W_d \triangle u_k \right\|_2^2 \tag{2.29}$$

where u_{ss} is the expected steady-states of the input which enable $r_{k+i} \rightarrow y_{k+i}$ asymptotically, r_{k+i} being the notional true output target. Nevertheless, the treatment of tracking has led to some minor changes to this popular index where the horizons n_y, n_u are large (or infinite) and it is impossible for the output prediction to reach the desired target while satisfying the constraints so that J could become unbounded.

A typical performance index can also be defined over the infinite horizon; for example

$$J = \sum_{i=0}^{\infty} \left\{ x_{k+i+1}^T Q x_{k+i+1} + u_{k+i}^T R u_{k+i} \right\} \tag{2.30}$$

This performance index provides a sensible definition of the optimum solutions for linear systems.

2.3.2 An alternative cost function

In practice, it is common for [101, 145] to define the disturbance estimate as the difference between the process output and model output ($d = y_p - y_m$). This disturbance estimate is used in combination with the desired target r in order to determine the steady-state values of the state (x_{ss}) and input (u_{ss}), which provide an offset-free tracking scenario. Embedding these disturbance estimates d_k and targets r_k into the control design is known as a tracking scenario.

There are circumstances where the typical performance index J (2.30) cannot be used due to the presence of model uncertainty. Therefore, an alternative performance index can be described that is conceptually the same as J but uses deviation variables rather than absolute variables (e.g. [145]).

$$J = \sum_{i=0}^{\infty} \left\{ (x_{k+i+1} - x_{ss})^T Q (x_{k+i+1} - x_{ss}) + (u_{k+i} - u_{ss})^T R (u_{k+i} - u_{ss}) \right\} \tag{2.31}$$

where Q, R are positive definite weighting matrices, u_{ss}, x_{ss} are the expected steady-states of the input and states which enable $y \to r_k$ asymptotically, with r_k being the desired target at sample k. Unbiased definitions of u_{ss}, x_{ss} and their linear dependence on the current disturbance estimate d_k and target r_k are well known in the literature (e.g. [101]) and subsection (2.1.2), and can be defined for suitable K_{xr}, K_{ur} as follows.

$$\begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} K_{xr} \\ K_{ur} \end{bmatrix} (r_k - d_k) \tag{2.32}$$

As part of this thesis, we will consider scenarios in which x_{ss} and u_{ss} violate the constraints.

The cost function of (2.31) can also be expressed in a compact form as:

$$J = \sum_{i=0}^{\infty} \left\{ (\hat{x}_{k+i+1})^T Q(\hat{x}_{k+i+1}) + (\hat{u}_{k+i})^T R(\hat{u}_{k+i}) \right\} \tag{2.33}$$

where $\hat{x} = x_{k+i+1} - x_{ss}$ and $\hat{u} = u_{k+i} - u_{ss}$ are the state and input deviations, respectively.

2.4 Constraints with finite horizons

Many processes contain constraints such as upper and lower limits on the input (input constraint), input rate (input rate constraint), and output (state constraint). Input constraints are usually referred to as hard constraints which must be satisfied, while output/state constraints may be referred to as soft constraints which should be satisfied if possible. More complex constraints can also be included without any modification being made to the concepts presented here. Assuming a linear model, these constraints can be captured as linear inequalities in the assumed future control moves and hence combined with the performance index to give a quadratic programming optimization which defines the control law. The key point is that the constraints should be described in terms of the degree of freedom (d.o.f); in this case, $\underset{\rightarrow k-1}{u}$.

This section will illustrate how to define these constraints in a standard form.

2.4.1 Input rate constraints

The input rate constraints at each sample can be expressed as:

$$\underline{\triangle u} \leq \triangle u_k \leq \overline{\triangle u} \tag{2.34}$$

The constraints on n_u future predicted input increments can be expressed in matrix inequality form as:

$$\underbrace{\begin{bmatrix} I \\ -I \end{bmatrix}}_{C_{\delta u}} \underbrace{\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+n_u-1} \end{bmatrix}}_{\underset{\rightarrow}{\triangle u}_{k-1}} \leq \underbrace{\begin{bmatrix} \overline{\triangle u} \\ \overline{\triangle u} \\ \vdots \\ -\underline{\triangle u} \\ \vdots \\ -\underline{\triangle u} \end{bmatrix}}_{\Delta U} \tag{2.35}$$

The control input can be described in terms of the future input increments as:

$$u_{k+i} = u_{k-1} + \triangle u_k + \triangle u_{k+1} + \cdots + \triangle u_{k+i}, \qquad i = 0, 1, ..., n_u - 1. \tag{2.36}$$

Building on equation (2.36), the future inputs can be described in a matrix form as:

$$\underbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+n_u-1} \end{bmatrix}}_{\underset{\rightarrow}{u}_{k-1}} = \underbrace{\begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ I & I & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & I & I & \cdots & I \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \triangle u_k \\ \triangle u_{k+1} \\ \vdots \\ \triangle u_{k+n_u-1} \end{bmatrix}}_{\underset{\rightarrow}{\triangle u}_{k-1}} + \underbrace{\begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}}_{L} u_{k-1} \tag{2.37}$$

Thus, the future input increments can be expressed in terms of the degrees of freedom (d.o.f) as:

$$\underset{\rightarrow}{\triangle u}_{k-1} = E_\Delta \underset{\rightarrow}{u}_{k-1} - E_\Delta L u_{k-1} \tag{2.38}$$

where $E_\Delta = (E)^{-1}$.

Combining equations (2.35) and (2.38) and with some algebra, as seen in [145], one can express the input rate constraints as:

$$\underbrace{C_{\delta u} E_\Delta}_{C_{\Delta u}} \underset{\rightarrow}{u}_{k-1} \leq \underbrace{\Delta U + C_{\delta u} E_\Delta L u_{k-1}}_{f_{\Delta u}} \tag{2.39}$$

or in a compact form:

$$C_{\Delta u} \underset{\rightarrow}{u}_{k-1} \leq f_{\Delta u} \tag{2.40}$$

where $C_{\Delta u}$ is a suitable matrix, $\underset{\rightarrow}{u}_{k-1}$ are the degrees of freedom (d.o.f), and $f_{\Delta u}$ is a vector of the limits.

2.4.2 Absolute input constraints

The upper and lower inputs on the input can be expressed as:

$$\underline{u} \leq u_k \leq \overline{u} \tag{2.41}$$

The constraints on n_u future predicted inputs can be also expressed in matrix form as:

$$\underbrace{\begin{bmatrix} I \\ -I \end{bmatrix}}_{C_u} \underbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+nu-1} \end{bmatrix}}_{\underset{\rightarrow}{u}_{k-1}} \leq \underbrace{\begin{bmatrix} \overline{u} \\ \overline{u} \\ \vdots \\ -\underline{u} \\ \vdots \\ -\underline{u} \end{bmatrix}}_{f_u} \tag{2.42}$$

or in a compact form:

$$C_u \underset{\rightarrow k-1}{u} \leq f_u \tag{2.43}$$

where C_u is a suitable matrix, $\underset{\rightarrow k-1}{u}$ are the degrees of freedom (d.o.f), and f_u is a vector of the limits.

2.4.3 State constraints

The state constraints at each sample can be described as:

$$\underline{x} \leq x_k \leq \overline{x} \tag{2.44}$$

Expressing the state constraints on n_u future predicted states using a matrix inequality yields:

$$\underbrace{\begin{bmatrix} I \\ -I \end{bmatrix}}_{C_x} \underbrace{\begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+nu-1} \end{bmatrix}}_{\underset{\rightarrow k}{x}} \leq \underbrace{\begin{bmatrix} \overline{x} \\ \overline{x} \\ \vdots \\ -\underline{x} \\ \vdots \\ -\underline{x} \end{bmatrix}}_{f_x} \tag{2.45}$$

$$C_x \underset{\rightarrow k}{x} \leq f_x \tag{2.46}$$

Substituting the state prediction (2.20) into (2.46) gives:

$$\underbrace{C_x P_x}_{M_x} x_k + \underbrace{C_x H_x}_{N_x} \underset{\rightarrow k-1}{u} \leq f_x \tag{2.47}$$

or in a compact form:

$$M_x x_k + N_x \underset{\rightarrow k-1}{u} \leq f_x \tag{2.48}$$

where M_x, N_x are suitable matrices and f_x is a vector of the limits.

In summary, the input rate, input and output constraints must be satisfied simultaneously; hence, equations (2.40, 2.43, 2.48) can be combined into a single set of linear inequalities of the form:

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ M_x \end{bmatrix}}_{\kappa} x_k + \underbrace{\begin{bmatrix} C_{\Delta u} \\ C_u \\ N_x \end{bmatrix}}_{\nu} \underset{\rightarrow k-1}{u} \leq \underbrace{\begin{bmatrix} f_{\Delta u} \\ f_u \\ f_x \end{bmatrix}}_{\tau} \tag{2.49}$$

or in a standard form which is suitable for QP optimisation:

$$\nu \underset{\rightarrow k-1}{u} \leq \tau - \kappa x_k \tag{2.50}$$

where κ, ν are the constant matrices, $\underset{\rightarrow k-1}{u}$ are the degrees of freedom (d.o.f), and τ is a vector of the limits.

2.5 Constraints with infinite horizons

The previous section demonstrated how input and state constraints can be tested over finite horizons. A key challenge is how these constraints can be tested over an infinite horizon while ensuring a finite computation load; that is, it is required for constraints' satisfaction to be captured over an infinite horizon, but using a finite number of inequalities. One method for achieving this was provided by [52] , using the concept of admissible sets. This section will briefly discuss this approach.

2.5.1 Maximal Admissible Set (MAS)

The admissible set algorithm is used for constraint handling to define the constraints over an infinite prediction horizon in terms of a finite set of inequalities linked to the d.o.f. It is based on an assumption that the predictions and constraints can be expressed as:

$$x_{k+1} = \Phi x_k, \qquad G x_k \leq f, \qquad \forall k \tag{2.51}$$

where Φ is a transition matrix. It is implicit that all of the eigenvalues of (Φ) lie strictly inside the unit circle.

The key concept is to test the constraints for the first n samples alone and then prove that the constraints must automatically be satisfied for all of the samples beyond that. If this is the case, then we can present constraints over an infinite horizon using only predictions over a horizon of n. A suitable value for n can be determined using the approach outlined in this section.

Remark 2.2 *Given a stability of* Φ, $\lim_{k\to\infty} x_k = 0$. *Assume* $Gx_0 < f$ *and for an asymptotic steady-state which is not on origin, a shift can be used to obtain an equivalent result.*

Algorithm 2.1 *The inequalities can be given for a specific horizon 'n' as:*

$$\underbrace{\begin{bmatrix} G \\ G\Phi \\ G\Phi^2 \\ \vdots \\ G\Phi^n \end{bmatrix}}_{F} x_k \leq \underbrace{\begin{bmatrix} f \\ f \\ f \\ f \\ \vdots \end{bmatrix}}_{t} \tag{2.52}$$

The key part of the algorithm is based on the following:

Assume that the transition matrix has stable properties, in which case we know that a suitable n must exist because

$$\lim_{k\to\infty} \Phi^k = 0. \tag{2.53}$$

1. *For a given 'n', try and find a value x_k which violates the constraints at '$n+1$' but satisfies constraints (2.52).*

2. *To do this, we substitute for state predictions in terms of the degrees of freedom $\underrightarrow{u}_{k-1}$ and use this optimisation for the upper limits:*

   For each row (i) of C, find the maximum value of state x_{max} as:

$$x_{max}(i) = \max_{x_k} e_i^T G \Phi^{n+1} x_k \quad \text{s.t.} \quad constraint\ of\ (2.52) \tag{2.54}$$

3. *If we cannot force a violation (i.e $e_i^T G x_{max} > f(i)$, then 'n' is sufficiently large; otherwise, increase 'n' and go to 1*

Details of how to express the predictions and constraints in the form of (2.51) for different MPC are included as required in later chapters.

2.5.2 Basic observations

This section has shown that:

1. Constraints satisfaction for prediction can be tested over an infinite horizon using a finite number of inequalities.

2. It is essential that the prediction dynamic is strictly convergent.

3. It is necessary for the asymptotic values not to lie on a boundary.

4. The set defined by these inequalities is often called a maximal admissible set or MAS [52].

2.6 Optimization and degrees of freedom (d.o.f) for finite horizons

Optimization is one of the main components of MPC algorithms. It describes the solution to the problem of optimizing the performance index J subject to system constraints. It is common to describe this optimization problem as a quadratic programming problem (QP), which can be solved to determine the desired optimal control input sequences.

Optimization for the finite horizon can be performed using equations (2.29, 2.50) and takes the general form:

$$\min_{\underset{\rightarrow k}{u}} J \qquad s.t. \qquad \text{constraint of (2.50)} \tag{2.55}$$

There are standard solvers for the QP methods which can provide an optimum solution. Popular methods are the active set method and the interior point methods [115, 170].

2.7 Dual mode (infinite horizon) MPC overview

The major weakness of the finite horizon algorithm is that using small control horizons n_u, n_y may result in poor performance and it is also known that it is difficult to guarantee asymptotic stability with finite horizons in MPC. This has led to the development of alternative infinite horizon algorithms to improve performance.

This section provides an overview of the most common infinite horizon approach; that is, the OMPC or dual-mode MPC approach [160, 153]. Although a dual-mode approach is more complex than a finite horizon because it includes terminal constraints, it has better properties.

2.7.1 Definition

The dual mode can be defined as a control strategy which has two modes. One mode is used when the system predictions are far away from the operating point (transient mode).

The second mode is used when the system predictions are close to the desired operating point (terminal mode). This is shown in Figure 2.1

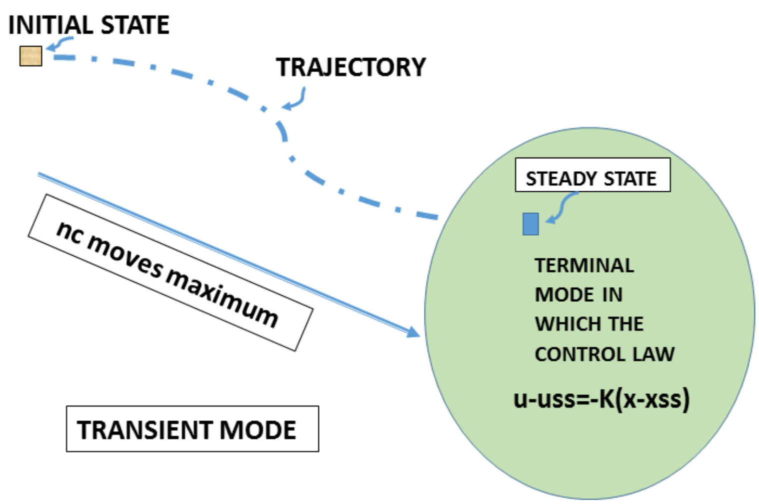

Figure 2.1: Basic dual-mode MPC structure

2.7.2 Degrees of freedom (d.o.f) with dual-mode MPC

It is common to define the degrees of freedom as the first n_c control increments (or moves); that is, $u_k, \cdots, u_{k+n_c-1}$. In the case of open-loop predictions, the first n_c control effort u_k are free and the remaining moves are given by a fixed feedback control law. Thus, one can define the predicted control values for the dual-mode as:

$$\begin{cases} u_k - u_{ss} = u_k - u_{ss}, & k < n_c \\ u_k - u_{ss} = -K(x_k - x_{ss}), & k \geq n_c \end{cases} \tag{2.56}$$

In this case, the degrees of freedom are $u_k, \quad k = 0, \cdots, n_c - 1$.

2.7.3 Dual mode and definition of the tail

The two main conditions which are sufficient to guarantee nominal stability are the use of an infinite horizon and the inclusion of the tail, which is defined next. A dual mode control uses infinite horizons and includes the tail. Hence, dual-mode algorithms can guarantee stability. Moreover, these allow a reduction in the number of d.o.f to be handled while still allowing the use of infinite input and output prediction horizons. This section defines 'the tail'.

The tail can be defined as the part of the optimum policy decided at the previous sample that has yet to be implemented.

Let the optimal predictions at sampling instant k be:

$$u_{k|k}, u_{k+1|k}, u_{k+2|k}, u_{k+3|k}, \cdots, u_{k+n_c-2|k}, u_{k+n_c-1|k}, u_{ss}, u_{ss}, u_{ss} \tag{2.57}$$

The tail from the previous sample is defined as:

$$u_{k|k-1}, u_{k+1|k-1}, u_{k+2|k-1}, u_{k+3|k-1}, \cdots, u_{k+n_c-1|k-1}, u_{ss}, u_{ss}, u_{ss}. \tag{2.58}$$

The tail is included in the current prediction, if one can choose the degrees of freedom at the current sample, such that $u_{k+i|k} = u_{k+i|k-1}$. $i = 0, 1, 2, .., n_c - 1$.

Remark 2.3 *It is obvious from (2.57) and (2.58) that the tail is included in the class of predictions of (2.56) and thus meets a fundamental requirement for expecting good a priori stability.*

2.8 Closed-Loop (CLP) Paradigm

The closed-loop paradigm was originally proposed in [55] and provided better numerical optimisation conditioning [75]. The CLP uses perturbations about the unconstrained optimal control law as the degrees of freedom (d.o.f). This formulation provides a good insight into the impact of constraints on performance and improves the conditioning of the optimisation.

This section introduces the closed-loop paradigm. Subsection 2.8.1 explores the structure of the CLP dual-mode while Subsection 2.8.2 explores the dual predictions and autonomous mode formulation. Subsection 2.8.3 describes the typical cost function used in dual-mode algorithms and Subsection 2.8.4 derives the dual-mode unconstrained control law. Subsection 2.8.5 describes how to define the constraints with the dual-mode while Subsection 2.8.6 defines the constrained control law for the dual-mode with constraints.

2.8.1 Dual mode prediction structure with the CLP

The closed-loop paradigm can be implemented as a dual-mode approach in which the initial state requires n_c moves to reach the terminal region.

For convenience, with infinite horizon algorithms, the d.o.f (or input parametrisation (2.56)) can be equivalently parametrised [146] as perturbations c_k about a nominal stabilising control law. Thus, unbiased state and input predictions in terms of deviation variables can be given as:

$$\begin{cases} x_{k+1} - x_{ss} = A(x_k - x_{ss}) + B(u_k - u_{ss}); \quad u_k - u_{ss} = -K(x_k - x_{ss}) + c_k, \quad k < n_c \\ x_{k+1} - x_{ss} = A(x_k - x_{ss}) + B(u_k - u_{ss}); \quad u_k - u_{ss} = -K(x_k - x_{ss}), \quad k \geq n_c \end{cases} \tag{2.59}$$

Let $\hat{x} = (x_k - x_{ss})$ and $\hat{u} = (u_k - u_{ss})$ be the state and input deviations respectively. One can then express the predictions of (2.59) in the form:

$$\begin{cases} \hat{x}_{k+1} = A\hat{x} + B\hat{u}; \quad \hat{u}_k = -K\hat{x} + c_k, \quad k < n_c \\ \hat{x}_{k+1} = A\hat{x} + B\hat{u}; \quad \hat{u}_k = -K\hat{x}, \quad k \geq n_c \end{cases} \tag{2.60}$$

.

The predicted state and input evolution is conveniently captured from equation (2.60). With $\Phi = A - BK$, a one-step ahead prediction model is:

$$\begin{cases} \hat{x}_{k+1} = \Phi\hat{x}_k + Bc_k; & \hat{u}_k = -K\hat{x}_k + c_k, \qquad k < n_c \\ \hat{x}_{k+1} = \Phi\hat{x}_k; & \hat{u}_k = -K\hat{x}_k, \qquad k \geq n_c \end{cases} \tag{2.61}$$

The model of (2.61) is commonly called dual-mode prediction because it contains two clearly distinct dynamics, one for transients (Transient mode $k < n_c$) and the other for asymptotic behaviour (Terminal mode $k \geq n_c$).

Now, it is obvious that the degrees of freedom are c_k.

2.8.2 CLP Predictions and autonomous model formulation

To deploy the concept of the MAS of (2.51) in dual-mode algorithms, it is convenient to capture both the transient and terminal modes of the predictions in a single mode as this enables us to deploy standard algebraic techniques, such as Lyapunov equations and global stability tests, more easily. A standard method for doing this is to construct an equivalent state-space model which incorporates the predictions of (2.61) by adding the d.o.f. $\underset{\rightarrow k}{c}$ as additional states.

The formulation which captures the two modes (2.61) in a single state space model can be given by the following augmented states:

$$Z_{k+1} = \Psi Z_k; \quad Z_k = [\hat{x}_k^T, \underset{\rightarrow k}{c}^T]^T; \quad \Psi = \begin{bmatrix} \Phi & [B, 0, .., 0] \\ 0 & D_c \end{bmatrix} \tag{2.62}$$

where:

$$\underset{\rightarrow k}{c} = \begin{bmatrix} c_k \\ c_{k+1} \\ \vdots \\ c_{k+n_c-1} \end{bmatrix}; \quad D_c = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{2.63}$$

Thus, the state deviation $\hat{x}_k$ can be described in terms of the autonomous model of equation (2.62), given that:

$$\hat{x}_k = \Gamma Z_k; \quad \Gamma = [I, 0, \cdots] \tag{2.64}$$

The input deviation $\hat{u}_k$ can also be described as:

$$\hat{u}_k = -K_Z Z_k; \quad K_Z = [-K, [1, 0, 0, 0 \cdots]] \tag{2.65}$$

The autonomous models (2.62) and (2.65) describe a state space representation of the dual-mode predictions.

2.8.3 The cost function of dual-mode MPC

In the most recent literature [128, 95], a typical infinite horizon algorithm cost function, assuming a non-zero target, is given by (2.31). This cost function can be optimized to determine the optimum value of the perturbation terms $\underset{\rightarrow k}{c}$. This subsection defines the cost function for a dual-mode algorithm in the absence of constraints.

A suitable cost function for the dual-mode MPC approach can be defined based on a mathematical illustration as follows.

As shown in equation (2.33), the cost function, which includes state and input deviation, is given by:

$$J = \sum_{i=0}^{\infty} \hat{x}_{k+i+1}^T Q \hat{x}_{k+i+1} + \hat{u}_{k+i}^T R \hat{u}_{k+i} \tag{2.66}$$

State and input deviations, which are described in terms of the autonomous model states (2.62), are described in (2.64, 2.65), respectively, given that:

$$\hat{x}_k = \Gamma Z_k; \quad \Gamma = [I, 0, \cdots] \tag{2.67}$$

$$\hat{u}_k = -K_z Z_k; \quad K_z = [-K, [I, 0, \cdots]] \tag{2.68}$$

Substituting (2.67, 2.68) into the performance index (2.66) gives a simple compact form:

$$J = \sum_{i=0}^{\infty} Z_{k+i}^T \Gamma^T Q \Gamma Z_{k+i} + Z_k^T K_z^T R K_z Z_k \tag{2.69}$$

Substituting (2.62) into the performance index of (2.69) gives:

$$J = \sum_{i=0}^{\infty} Z_k^T [\Psi^T \Gamma^T Q \Gamma \Psi + K_z^T R K_z] Z_k \tag{2.70}$$

The cost function of (2.70) can be described in a compact form as:

$$J = Z_k^T [\sum_{i=0}^{\infty} (\Psi^i)^T \Psi^T \Gamma^T Q \Gamma \Psi + K_z^T R K_z \Psi^i] Z_k \tag{2.71}$$

Assuming that:

$S_z = \sum_{i=0}^{\infty} (\Psi^i)^T \Psi^T \Gamma^T Q \Gamma \Psi + K_z^T R K_z \Psi^i$ and $W = \Psi^T \Gamma^T Q \Gamma \Psi + K_z^T R K_z$.

Thus, the cost function of J of (2.71) can be expressed in a simplified form [145]:

$$J = Z_k^T S_z Z_k; \tag{2.72}$$

Now, the term S_z can be expressed in terms of the term W as:

$$S_z = \sum_{i=0}^{\infty} (\Psi^i)^T W \Psi^i \tag{2.73}$$

Splitting the equation (2.73) into two parts, one part is the first component of the sum infinite of the number of terms and the other part is the rest of the sum infinite, giving:

$$S_z = W + \sum_{i=1}^{\infty} (\Psi^i)^T W \Psi^i \tag{2.74}$$

Expressing the equation (2.74) in a compact form yields:

$$S_z - W = \sum_{i=1}^{\infty} (\Psi^i)^T W \Psi^i \tag{2.75}$$

Multiplying the equation (2.73) by Ψ and Ψ^T gives:

$$\Psi^T S_z \Psi = \sum_{i=0}^{\infty} (\Psi)^T (\Psi^i)^T W \Psi^i \Psi \tag{2.76}$$

which is equivalent to:

$$\Psi^T S_z \Psi = \sum_{i=1}^{\infty} (\Psi^i)^T W \Psi^i \tag{2.77}$$

Thus, it is straightforward to define the Lyapunonv identity based on equations (2.77) and (2.75) as follows.

$$\Psi^T S_z \Psi = S_z - W \tag{2.78}$$

S_z is the sum of the infinite number of terms and can be solved using a simple matrix form in terms of W and Ψ using some linear equalities given that:

$$S_z = \begin{bmatrix} S_x & S_{xc} \\ S_{xc}^T & S_c \end{bmatrix} \tag{2.79}$$

Hence, the cost function (2.72) can be expressed as:

$$J = \begin{bmatrix} \hat{x}_k \\ \underrightarrow{c}_k \end{bmatrix}^T S_z \begin{bmatrix} \hat{x}_k \\ \underrightarrow{c}_k \end{bmatrix}; \qquad S_z = \begin{bmatrix} S_x & S_{xc} \\ S_{xc}^T & S_c \end{bmatrix} \tag{2.80}$$

Now, the compact cost function can be expanded to:

$$J = \hat{x}_k^T S_x \hat{x}_k + 2\hat{x}^T S_{cx} \underset{\rightarrow k}{c} + \underset{\rightarrow k}{c}^T S_c \underset{\rightarrow k}{c} \tag{2.81}$$

2.8.4 Optimisation and the unconstrained dual-mode control law

Minimising the performance index (2.81) w.r.t to the perturbation $\underset{\rightarrow k}{c}$ to determine the optimum solution requires the derivatives of J to be zero. This can be given as:

$$\frac{dJ}{d\underset{\rightarrow k}{c}} = 2S_c \underset{\rightarrow k}{c} + 2S_{xc}^T \hat{x}_k = 0 \tag{2.82}$$

Consequently:

$$\underset{\rightarrow k}{c} = -[S_c]^{-1} S_{cx}^T \hat{x}_k \tag{2.83}$$

The perturbation c_k is the first element of $\underset{\rightarrow k}{c}$ and can be expressed as:

$$c_k = -e_I^T [S_c]^{-1} S_{cx}^T \hat{x}_k; \qquad e_I^T = [I, 0, 0 \cdots] \tag{2.84}$$

The unconstrained control law can be defined as:

$$\hat{u}_k = -K\hat{x}_k + c_k \tag{2.85}$$

Substituting (2.84) into (2.85), we get the unconstrained control law:

$$\hat{u}_k = -\underbrace{[K - [S_c]^{-1} S_{cx}^T]}_{\hat{K}} \hat{x}_k; \quad i < n_c \tag{2.86}$$

Or, in a compact form:

$$\hat{u}_k = -\hat{K}\hat{x}_k \tag{2.87}$$

Remark 2.4 *By definition, the optimal behaviour is given from $\hat{u}_k = -K\hat{x}_k$; therefore, by definition, the optimal value of c_k must be zero, therefore $S_{xc} = 0$.*

Although this means the optimal $\underset{\rightarrow k}{c} = 0$ in unconstrained case, this is laying important ground work for constraint handling, which will be discussed next.

2.8.5 Including the constraints into dual-mode MPC

This subsection will show how to define the constraints over the infinite horizon for the dual-mode.

Considering the predictions model (2.11), the input constraint at each sample can be expressed in terms of the steady-state values as:

$$\underline{u} \leq \hat{u}_k + u_{ss} \leq \overline{u} \tag{2.88}$$

Expressing input constraints in terms of the augmented state and input (2.62, 2.65) gives:

$$\begin{bmatrix} -K_Z \\ K_Z \end{bmatrix} Z_k \leq \begin{bmatrix} \overline{u} \\ -\underline{u} \end{bmatrix} + \begin{bmatrix} -u_{ss} \\ u_{ss} \end{bmatrix} \qquad \forall k \tag{2.89}$$

The state constraint (2.44) at each sample can be also expressed in terms of the steady-state values as:

$$C_x(\hat{x} + x_{ss}) \leq \overline{x}, \quad \forall k \tag{2.90}$$

Thus, the state constraints can be expressed in terms of the augmented state, which gives:

$$\begin{bmatrix} C_x & 0 \end{bmatrix} Z_k \leq \overline{x} - C_x x_{ss} \tag{2.91}$$

$$\underbrace{\begin{bmatrix} -K_Z & \\ K_Z & \\ C_x & 0 \end{bmatrix}}_{G} Z_k \leq \underbrace{\begin{bmatrix} \overline{u} \\ -\underline{u} \\ \overline{x} \end{bmatrix} + \begin{bmatrix} -u_{ss} \\ u_{ss} \\ -C_x x_{ss} \end{bmatrix}}_{f} \tag{2.92}$$

The steady-state values x_{ss} and u_{ss} in equation (2.92), can be described in terms of the desired target and disturbance estimate gives:

$$\underbrace{\begin{bmatrix} -K_Z & \\ K_Z & \\ C_x & 0 \end{bmatrix}}_{G} Z_k \leq \underbrace{\begin{bmatrix} \overline{u} \\ -\underline{u} \\ \overline{x} \end{bmatrix} + \begin{bmatrix} -K_{ur} \\ K_{ur} \\ -C_x K_{xr} \end{bmatrix} (r_k - d_k),}_{f} \quad \forall k. \tag{2.93}$$

$$GZ_k \leq f \tag{2.94}$$

Lemma 2.1 *Constraint handling with dual-mode MPC can be described in a standard form for the MAS.*

Proof: Using equations (2.62) and (2.94) is equivalent to equation (2.51); therefore, using the theory of the admissible sets [52], one can define an admissible set in the form:

$$\underbrace{\begin{bmatrix} M & N \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} \hat{x}_k \\ \underset{\rightarrow k}{c} \end{bmatrix}}_{Z_k} \leq t \tag{2.95}$$

or, in a compact form:

$$S_{MCAS} = \left\{ \hat{x} : \exists \underset{\rightarrow k}{c} \;\; s.t \;\; M\hat{x}_k + N \underset{\rightarrow k}{c} \leq t \right\} \tag{2.96}$$

where M and N are suitable matrices and t is a vector of the limits.

The corresponding set is called th maximal controlled admissible set ($MCAS$), since it includes the d.o.f within c_k as well as the initial states.

Remark 2.5 *The inequalities in (2.93) depend upon the expected steady-state and this may change with both the target and disturbance estimate. This would be difficult for typical MAS algorithms [52] because they assume fixed targets.*

One challenge in this thesis is how to construct the MCAS for time-varying targets. This will be discussed later, in Chapter 4.

2.8.6 Optimisation and the constrained dual-mode control law

Subsection 2.8.4 demonstrates the derivation of the control law for unconstrained systems for the regulation case. For constrained systems, a typical algorithm minimises a performance index (J) subject to the corresponding constraints S_{MCAS} and, using an input trajectory/d.o.f. of the optimised $\underrightarrow{c}_k$, only the first value c_k is deployed and the optimisation is repeated for every sample.

It is shown in [145] that the cost function of (2.81) can be reduced to a simple form as:

$$\min_{\underrightarrow{c}_k} J = \underrightarrow{c}_k^T S_c \underrightarrow{c}_k \tag{2.97}$$

Thus, the dual-mode optimisation can be given by:

$$\min_{\underrightarrow{c}_k} J = \underrightarrow{c}_k^T S_c \underrightarrow{c}_k \qquad s.t. \qquad M\hat{x}_k + N\underrightarrow{c}_k \leq t \tag{2.98}$$

The constrained control law is then defined as:

$$\hat{u}_k = -K\hat{x}_k + c_k. \tag{2.99}$$

2.9 Feed-Forward (FF) Compensator

One purported advantage of MPC is that it includes information about future targets, $\underrightarrow{r}$ [145]. This section will show that the MPC control law includes future information about set point changes and it is expected that including this future information in the optimization of the predicted performance index will improve tracking. However, the inappropriate use of future information may result in poor performance.

For simplicity, we will consider that the augmented state space model of (2.3) with the corresponding unconstrained control law is given by:

$$u_k = -K(x_k - x_{ss}) + u_{ss} \tag{2.100}$$

Substituting the x_{ss} and u_{ss} of (2.32) into (2.100) gives:

$$u_k = -\tilde{K}_k \mathbf{x}_k + P_r r_k \tag{2.101}$$

where $\tilde{K} = K + [KK_{xr} + K_{ur}]$, $P_r = KK_{xr} + K_{ur}$, $\mathbf{x}_k = \begin{bmatrix} x_k \\ d_k \end{bmatrix}$ is the augmented state.

Remark 2.6 *The control law (2.101) includes a feed-forward term $P_r r$. Although just on current target r_{k+1}.*

Ideally, one should use more information from the target. i.e $r_{k+1}, r_{k+2}, \ldots, r_{k+n_y}$. Therefore, one can define a more generic feed-forward term as:

$$P_r \underrightarrow{r}_{k+1} = \begin{bmatrix} P_1 & P_2 & \cdots & P_{n_y} \end{bmatrix} \begin{bmatrix} r_{k+1} \\ r_{k+2} \\ \vdots \\ r_{k+n_y} \end{bmatrix} \tag{2.102}$$

where $\underrightarrow{r}_{k+1}$ is the future target and P_r is a feed-forward term.

It is shown from equations ((2.101), (2.102)) that the future targets have a direct impact on the future current control move.

In many MPC algorithms, a default value of P_r, can be defined, but this value is often poor, as will be discussed further in Chapter 4.

2.10 Multi-Parametric Quadratic Programming (mp-QP)

Model predictive control can successfully handle constrained multi-variable control problems. The solution to the MPC control problem is to compute the control input u_k at each sampling time by optimising a quadratic performance index over a finite horizon. This is known as a quadratic programming (QP) problem [39]. Solving the QP problem is based on the current state measurement x_k and repeated optimisation at each sampling time. This may result in a significant on-line computation burden hence limiting the application of the MPC to slow or small systems. Therefore, an alternative method, such as Multi-Parametric Quadratic Programming (mp-QP) [67, 15, 101], has been presented to reduce the on-line computational load. Those mp-QP methods can be used to reduce the computational load by moving as much of the constraint handling as possible to off-line computation rather than on-line optimisation. It is shown in [15] that the MPC optimisation problem can be reformulated as an mp-QP problem.

The standard mp-QP problems take the form:

$$V(x) = \min_{U} \ \frac{1}{2}U^T H U + x^T F^T U \qquad s.t. \qquad GU \le W + Ex \tag{2.103}$$

where the column vector $U = \left[u_0^T, u_1^T, \cdots, u_{N-1}^T\right]^T \in R^m$ is denoted as the optimisation vector, $x \in R^n$ is a parameter and H, G, W, E and F are the real matrices which can be obtained from Q and R.

It is useful to define:

$$z = U + H^{-1}F^T x \tag{2.104}$$

$z \in R^s$, and to reformulate the problem of (2.103) as:

$$V_z(x) = \min_{z} \ \frac{1}{2} z^T H z \qquad s.t. \qquad Gz \leq W + Sx \tag{2.105}$$

where $S = E + GH^{-1}F^T$ and $V_z(x) = V_x - \frac{1}{2}FH^{-1}F^T x$

The main objective of the mp-QP algorithm is to solve the QP off-line for all x to determine the MPC control law u(x) explicitly, assuming that $H = H^T \succ 0$. It is also assumed that all constraints on x are included in (2.50), without any redundant constraints [163].

The solution of the QP problem (2.105) can be defined using an explicit piecewise affine MPC control law as:

$$u(x) = \begin{cases} F_1 x + g_1 & \\ \vdots & \textbf{if} \quad \vdots \\ F_m x + g_m & \end{cases} \begin{matrix} H_1 x \leq K_1 \\ \\ H_m x \leq K_m \end{matrix} \tag{2.106}$$

It is shown in the literature [15] that the linear MPC controller is a continuous piecewise affine function of the state x . Equation (2.106) is the mp-QP solution.

2.11 Summary of the basic results

This section summarises some of the key points that can be extracted from the above MPC background representation.

1. It is convenient to use unbiased predictions in MPC algorithms.

2. Constraints satisfaction for input and output predictions can be tested over both finite and infinite horizons, using a finite number of inequalities.

3. The closed-loop paradigm uses perturbations to the unconstrained optimal control law as the degrees of freedom. This provides a useful insight into the impact of constraints

on performance and improves the conditioning of the optimisation.

4. The CLP MPC control law in the unconstrained state space case is equivalent to a fixed state feedback.

5. The computation of the MPC control law using CLP reduces the solution of a quadratic programming problem.

6. The constraint equations are affined in the d.o.f, the current state, target and the limits.

7. A MCAS can be constructed for dual-mode MPC algorithms. This set ensures that the predictions satisfy the constraints over an infinite horizon using a finite number of inequalities.

Chapter 3

LITERATURE REVIEW

There are some useful MPC books and publications, such as ([145, 19, 90, 96, 167, 73]) can be found, showing various interesting insights into predictive control. On the other hand, there is much other literature on this field. This chapter presents a literature review of the model predictive control approaches with an emphasis on efficient algorithms and components which would be useful for real application. Section 3.1 describes some of the industrial processes which deploy different strategies for the predictive control approach. Section 3.2 introduces several works on infinite horizon predictive control. Section 3.3 reviews the feasibility and stability issues in the constrained predictive control approaches, including two common strategies for handling these issues. Section 3.4 presents several works on the offset-free control methods. Section 3.5 describes the preview and feed-forward options in the predictive control approach. Section 3.6 presents an overview of the MPC algorithms which handle uncertain systems, while Section 3.7 describes an optimisation method (mp-QP) to be implemented in MPC for fast applications. Finally, a list of observations is provided in Section 3.8.

3.1 Model predictive control with industrial process control

Applications of model predictive control (MPC) to the process industry were presented in [19], including a brief history of industrial MPC technology. Furthermore, various MPC approaches were examined and analysed, demonstrating the potential of the MPC control. A good survey of the application in the industry can be also found in [119, 120]. In their survey, the authors showed that predictive control has been successfully applied to several industrial processes such as food processing, automotive, aerospace and chemical processes with low sampling requirements. They also found that the majority of the non-linear MPC applications were found in the chemicals, air/gas, and polymer industries. However, recently, it has been demonstrated that model predictive control can be implemented to control other processes such as electrical machines and drives [32, 171]. In this section, an overview of the most commonly used three strategies of model predictive control, with a brief description of some of their implementation regarding process control, is provided, showing their efficacy and success in the industry.

3.1.1 Dynamic Matrix Control (DMC/QDMC)

The Dynamic Matrix Control (DMC) technique [33] is one of the most popular techniques in industry. It uses a step response model of the process to create the predictions. The objective of the DMC controller is to drive the output as close to the set point as possible using a least squares problem with a performance index over a finite prediction horizon. Several works have been found in the literature that demonstrate the success of the implementation of the DMC in industrial processes as described in this subsection.

The details of the unconstrained multi-variable control algorithms (DMC) are presented in [33]. Cutler and Ramaker [33] demonstrated improved control quality for furnace temperature control application when the DMC algorithm is implemented. A minor modification was made to the DMC algorithm by Cutler and Ramaker [33]. This modification was based on considering the DMC algorithm as a quadratic program (QP), in which the input and output constraints appear explicitly. This modified algorithm was denoted the QDMC algorithm.

An application of DMC technology to an FCCU reactor/regenerator control, in which the algorithm was modified to handle non-linearity and constraints was presented in [118]. The author modified the algorithm to prevent absolute input constraint violation.

The advantages of the DMC algorithm and indeed its capability with regard to constraint handling were explored in [49]. It was shown that the DMC approach can be applied in both linear and non-linear systems successfully, particularly in the petrochemical industry, with multi-variable processes [120, 64].

Garcia and Morshedi [48] presented results from a pyrolysis furnace application and adopted the QDMC algorithm to control stream temperature in the furnace. The authors demonstrated that the QDMC algorithm provides a systematic way of handling input and output constraints, thereby providing good results for such an application.

The design of a DMC to control the outlet temperature of a water heater was introduced in [19]. It was shown in this work that the DMC controller provides a fast response to set point change and is able to reject the applied measurable disturbance. The authors also discussed the application of the DMC to a multi-variable process such as a chemical jacket reactor, and demonstrated that the output follows the temperature reference satisfactorily in the case of a temperature reference change.

Summary: All of the industrial applications described in this subsection show the success of the DMC in dealing with various industrial processes. However, the DMC does not perform well in the case of constraints, except for the QDMC, which provides a systematic methods for implementing input and output constraints. Moreover, there was no clear strategy for dealing with an infeasible solution.

3.1.2 Predictive Functional Control (PFC)

Predictive Functional Control (PFC) is an MPC formulation that is used on a wide range of applications but often differs from the traditional DMC. The key objective of the PFC which distinguishes it from the MPC algorithms, is its focus on the simplicity of the concept as well as the coding. Therefore, it can be implemented successfully for many industrial processes [133]. This controller uses a simplified optimisation procedure by only taking specific coincidence points rather than minimising a cost function, providing a faster calculation of the control input. Moreover, the algorithm uses basis functions in order to construct the control signal which permits the controller to track different targets. Evidence of the success of PFC's application in industry will be provided in this subsection as follows.

The PFC control technique was applied in [77] to an elastic industrial robot. Kuntzee et al [77] demonstrated that the controller provided both excellent tracking behaviour and robustness with respect to parameter variations.

The applications of PFC to several process control such as the thickness of a cold rolling mill and the water level of the Rhone River, together with the position of a fast robot, were described in [31]. The obtained results showed the success of the PFC implementation with regard to such process control.

The PFC's application to a missile control system was presented in [65] to study the robustness and tracking performance. Jianbo et al [65] showed that the PFC controller provided a simple calculation, strong robustness, disturbance attenuation and high control precision.

The implementation of the PFC approach to a chemical batch jacketed reactor to control the outlet temperature, was applied by Bouhenchir et al [18]. The authors demonstrated that the PFC technique can be considered a suitable solution for controlling the batch reactors temperature, and that the developed PFC framework can be applied to control industrial batch reactors.

Maalouf [89] demonstrated the possibility of using the PFC algorithm to control an $H4$ parallel robot and showed that the controller provides an equilibrium point for the closed-loop system and track at the same time reference.

In [58], a PFC algorithm was installed successfully for the control of two distillation columns and a reactor at a petro-chemical plant. It is shown that the control behaviour was robust, reliable, and suitable for such processes.

Recently, the PFC algorithm has been implemented in [82] to control the permanent magnetic synchronous motor servo system. The author verified that the controller is effective and can ensure closed-loop optimal performance.

More recently, an PFC strategy based on a novel formulation of the state space model in the presence of plant model mismatch was introduced in [174]. The strategy has been applied to the outlet radiation temperature control of the coke furnace. Zhang et al [174] showed that the PFC controller successfully maintained the required set-point, thereby providing satisfactory tracking performance and robustness.

Summary: The review presented in this subsection shows that the PFC can provide good performance particularly for fast processes due to its fast control moves calculations. However, except for the recent work [174], applications of PFC have largely been restricted to single input single output and two input two output processes [131] because the simplicity of the control definition will not in general handle interaction effectively.

3.1.3 Generalized Predictive Control (GPC)

Generalized predictive control (GPC) [26] is a popular predictive control approach that can handle industrial processes successfully [27]. The GPC strategy is based on a description of the system by a Controlled Auto-Regressive Integrated Moving Average (CARIMA) model and a cost function which penalises predicted deviations from a constant reference. The cost function is then minimised to obtain the optimal control law. This subsection will focus on reviewing some of the existing work in this approach showing it's application within process control.

An application of GPC to some industrial processes such as a cement mill, a spray-drying tower, and compliant robot arms was introduced in [27], showing the efficacy of the GPC approach for a self-tuning control of industrial processes. It was also shown that the GPC approach outperformed a fully tuned PID control on an industrial processes.

To design an effective MPC approach for systems with constraints, Tsang and Clarke [164] proposed a GPC algorithm which handles constraints and improves the closed-loop performance. The proposed algorithm was implemented on a single-link flexible robot arm. It was shown that this algorithm provides good performance and is computationally acceptable.

MPC algorithms can be implemented for different systems and provide good results. For example, in [30], in which a novel algorithm was proposed, it was demonstrated that a GPC approach is suitable for controlling complex plants, such as unstable systems. Furthermore, it was shown that the algorithms can be used for several industrial processes.

In [142], a design study on a model of a boiler-turbine system using the GPC algorithm was presented. The controller provided set point tracking with minimal control action. Rossiter et al [142] demonstrated that the GPC is efficient for use with multi-variable systems.

Regarding to the stability issue, a novel MPC framework that uses the idea of the GPC was proposed in [74]. This framework guarantees system stability by ensuring that the optimal predicted cost is a Lyapunov function for the closed-loop system and reduces the computational burden associated with a computer-based implementation of the method.

In [104], the GPC was applied to a linear gas turbine model of power generation. It was shown that the GPC provides good results for the constrained case.

Camacho and Bordons [19], discussed the success of GPC implementation in various industrial SISO processes, such as a simple furnace and an evaporator, as well as MIMO processes, such as a stirred tank reactor, a distillation column and an air compressor. It was shown in [19] that the GPC algorithms can provide good behaviour, in which the output variables reach their set point in a very short time and reduce the interaction between the controlled variables in the case of MIMO systems.

Recently, in [46], the GPC strategy has been applied to a complex industrial process, that is polyvinyl chloride (PVC) polymerisation reactors, to control their non-linear temperature. It has been demonstrated in this work that the GPC controller improves the quality of the PVC product and reduces the production costs.

More recently, the application of the GPC approach to the hydrocarbon temperature of a process control model was described in [124]. It was shown from the result obtained from this work was that the GPC algorithm provides good, accurate tracking performance in the presence of disturbance.

A novel MPC formulation which is linked to GPC, called constrained receding horizon control (CRHPC), was developed in [28] for stabilising unstable, non-minimum phase, and dead-time plants. It was demonstrated that this framework provides good behaviour even with unobservable systems. This technique proved that the CRHPC approach can outperform the existing methods such as generalised minimum variance and pole placement as well as a fully tuned self-tuning application.

Summary: As shown in this subsection, many frameworks have studied the GPC model predictive control approaches for various systems in industry and these approaches are often effective in many scenarios. However, there is one notable issue that has yet to be considered, specifically, the design of the feed-forward component. It is shown in [145] that the default feed-forward from a GPC type of algorithm is often poor due to the mismatch between the predicted and desired behaviour that may result from the optimisation problem. Therefore, there is a need for a mechanism of how to improve the design of the feed-forward.

3.1.4 Summary

The industrial applications described in this section show that predictive control is a successful approach to various industrial processes. This is logical since it possesses several characteristics which lead to this success such as the systematic handling of constraints, being easy to explain and understand, and the availability of their software tools [90]. In particular, the DMC and GPC algorithms are based on a finite prediction horizon and are good enough to deal with most industrial processes while PFC tends to be restricted to processes with low computations. However, a severe problem with GPC is that it does not have a feasibility and stability guarantee, which is linked to the use of infinite horizons. Therefore, there is a need to use infinite horizons in advanced MPC process applications.

3.2 Infinite horizon predictive control algorithms

Regarding the recent development of predictive control, the literature shows that infinite horizon predictive control algorithms can guarantee the convergence of the predicted trajectories. This section will review some of the existing work on these algorithms to explore the benefits of using an infinite horizon in predictive control.

An original approach to infinite horizon predictive control was presented in [128]. It was demonstrated in this work that the infinite horizon in predictive control can guarantee stability for both unconstrained and constrained cases.

In [101], a unified framework of the existing MPC concepts based on stabilizing, infinite horizon, linear quadratic regulator was discussed. The framework offers the flexibility to handle various systems, such as non-square systems, noisy inputs and outputs, non-zero input and output and state disturbances. Furthermore, this framework removes the steady-state offset by including the integral action and state estimation within the controller.

Stability with the infinite horizon algorithms was discussed in [138], by deploying necessary and sufficient conditions for the convergence of the predicted trajectories. Also an alternative means of computing the implied infinite horizon GPC cost was presented. In [22], an alternative infinite horizon predictive control algorithm was introduced. This algorithm also guarantees closed-loop stability for both stable and unstable systems under input constraints.

Maciejowski [90] examined the use of infinite horizons in predictive control and showed that this provides stability. The author also demonstrated that stability can be achieved for both constrained stable and unstable plants.

The advantages of infinite horizon optimal control were demonstrated in [95], based on a discussion of the stability and optimality of the infinite horizon predictive control for both linear and non-linear systems with constraints. It was found from this discussion that several factors are useful in developing stabilizing model predictive controllers and that these controllers can be stabilized under sufficient conditions for both nominal and robust cases.

In terms of work on infinite horizon predictive control, one challenge is the computational load that results from the optimisation of the performance index over infinite horizons. This challenge has been tackled using an appropriate approach, such as the dual-mode MPC approach [153, 145], which allows a reduction to a quadratic program with a finite number of degrees of freedom and a finite number of constraints.

A dual-mode MPC approach in the form of a constrained linear quadratic regulator (LQR) was presented in [153]. Scokaert and Rawlings [153] showed that the presented regulator improves performance and removes the parameter which is required for tuning the controller. Moreover, the LQR regulator can remove the mismatch between open-loop and closed-loop nominal behaviour. Scokaert and Rawlings observed that the algorithm can be applied to any stable system.

Rossiter [145] mentioned that the dual-mode paradigm is computationally tractable and relatively simple to implement. Rossiter [145] also stated that the significant issue in dual-mode MPC design is the trade-off between the terminal set and good performance. However, this issue has been tackled by a closed-loop dual-mode paradigm.

Recently, the dual-mode MPC paradigm based on the concept of the invariant set, was described in greater detail in [73]. The authors introduced an algorithm which guarantees an asymptotic convergence of the state and Lypunov stability and showed that the introduced algorithm is recursively feasible.

Summary: All infinite horizon algorithms described in this section exhibited stability properties with satisfactory performance for the applied control schemes. In particular the optimal predictive control or dual-mode approach, which is deemed to be rigorous and can guarantee feasibility, hence enables guarantees of nominal and robust closed-loop stability for the controller [145]. Therefore, the focus of this thesis will be on dual-mode or OMPC predictive control approaches as the basic tools used in this research.

3.3 Feasibility and stability within constrained MPC tracking

It is common to embed several future values of the desired target, r, and disturbance estimate d, in the optimisation control problem. Applying these data to the optimisation problem within MPC is known as a MPC tracking problem. It is striking that MPC tracking solutions have largely been considered as regulating problems, about a steady-state operating point, rather than full tracking problems. This section demonstrates the recent work on important issues in predictive control, such as the feasibility and stability guarantee within a MPC tracking problem. The following subsection presents a brief description of the feasibility and stability problems within the tracking scenario, followed by a comprehensive literature review covering the discussion on the solution to the feasibility and stability problems of the associated MPC optimizations.

3.3.1 Feasibility and stability problems

One of the challenges associated with a tracking scenario is the need to ensure feasibility, which can guarantee that the class of predictions available to the MPC algorithm can indeed satisfy all of the constraints simultaneously. However, even putting aside issues linked to model uncertainty, feasibility can easily be lost during rapid or large set point changes and disturbance changes, both of which have a strong impact on the terminal constraints [86, 145]. Consequently, a strong link exists between set point tracking and feasibility; the feasibility of the controller may be lost and the controller ill-defined, or not defined at all [126], in the case of certain set point changes.

A convenient and essential component for enabling stability guarantees of MPC algorithms is to ensure feasibility; that is, to ensure the *class of predictions* over which an optimisation is being performed, and to include at least one which is able to satisfy all of the constraints, including the terminal constraint. That is, for a suitable underlying MPC approach such as dual-mode [153, 146], a feasibility guarantee is often sufficient to enable a simple guarantee of nominal (and at times robust) closed-loop stability for the controller. However, it is shown in [19, 90] that the inclusion of hard constraints may result in an infeasible optimisation problem and lead to closed-loop instability even when the unconstrained optimisation problem would lead to a system that is stable in a closed-loop.

Two common strategies have been applied to address the problem of infeasibility and instability during large disturbances, plant model mismatch, and rapid set point changes. These strategies are denoted in the literature as output or state soft constraints [90, 1] and the set point management [145, 1]. An overview of some of these strategies is presented in the following subsections.

3.3.2 Soft constraints with predictive control

Soft constraints [90, 154] is a technique that allows state or output constraint violations but, an additional slack variable that penalises the constraint violation is included in the cost function of the predictive control. This is considered one of the most popular techniques for addressing infeasibility and instability problems. Several algorithms for this technique may be found in literature. In the following subsections, an overview of some of these algorithms will be presented.

Quadratic (l_2) penalty formulation methods

Zafiriou and Chiou [172] presented an output soft constrained MPC formulation for a SISO system to address the problem of stability induced by step output disturbance. In this work output constraints are softened by adding a squared penalty with a softening weight parameter to the cost function. At this point, Zafiriou and Chiou [172] provided a technique for computing the largest softening weight that will not cause any stability problems. Under certain assumptions, the softening weight can be obtained numerically or from the Nyquist plot. Moreover, the author reported that this technique can be extended to the MIMO case as well as the case of modelling error.

An alternative MPC formulation, that penalizes a square of the peak violation over the horizon, weighted by a constant weight, in the cost function was presented in [175]. The authors studied the stability and showed that the global asymptotic stability of linear time-invariant discrete time systems with mixed hard and soft constraints could be guaranteed with poles inside the closed unit disc subject to hard input constraints and soft output constraints. The advantage of this formulation is that it is computationally cheap and requires the solution to a single QP problem. However, it provides poor performance and tuning difficulties. Moreover, for some cases such as non-minimum phase plants, this becomes a multi-objective minimization as discussed in [154].

Rossiter and Kouvaritakis [140] presented a solution which enforces soft constraints by minimizing the maximum value of the state constraint violation. This value is minimized by using a different optimization problem that is solving the constrained QP as an iteration of weighted least square optimization problems.

Linear (l_1 / l_∞) norm penalty function methods

It is desired to soften the state constraints only when the optimization is infeasible. This can be achieved by using the concept of an exact penalty function [43]. This idea is exploited by several authors to formulate an exact soft constrained MPC [35, 154, 69].

Kerrigan and Maciejowski [69] introduced a method for calculating a lower limit for the weight of the constraint violation penalty. In this work, Kerrigan and Maciejowski [69] showed that the lower limit can be calculated by means of a linear programs' problem solution based on the KKT conditions. The authors also demonstrated that, in order to guarantee that the solution with soft constraints is equal to that with hard constraints for all feasible considered conditions, the weight of the constraint violation which is used in the soft constrained cost function must exceed the maximum norm of the Lagrange multiplier. Kerrigan and Maciejowski [69] reported that this method can guarantee that the lower bound has been found. However, this can prove an intractable problem [63].

Maciejowski [90] discussed how to solve the infeasibility problem induced by the presence of modelling error and disturbances, using both the l_1 norm and l_∞ norm penalty function approaches for the output soft constraints. With regard to modelling error, the author demonstrated that the inexactness of the penalty function leads to some constraint being violated, whereas other constraints are not, despite the infeasibility induced by the modelling error; hence, it is difficult to predict the circumstances in which infeasibility will be encountered. In the case of disturbances, the author also showed the efficacy of softening constraints and that the inexactness can create a useful compromise between the hardness and sponginess of set point specification.

Wills and Heath [169] argued that choosing a large linear term in the cost function in order to obtain exact penalty function may not be useful for system dynamics. Therefore, they proposed a two-stage process which includes the exact penalty method as a special case, such that the feasibility of the MPC optimization problem is determined first, and then the MPC problem depending on the new information is solved. This development is discussed based on the classical exterior/interior point framework [42], which allows for an intuitive tuning procedure. They have shown that this proposal can improve performance by using a heuristic choice of the soft constraint penalty scaling factor.

Hovd [63] proposed an alternative method for ensuring exact soft constraints by computing a sufficient maximum value of the l_1 norm and l_∞ norm of the Lagrange multiplier for the standard QP problem using a multi-level programming.

A new MPC formulation for soft constraints was proposed in [173] for tracking linear constrained systems. The proposed formulation can guarantee stability for both stable and unstable systems. The concept is to relax the terminal constraints using an enlarged terminal set and to soften the other constraints by using both the quadratic l_2 and linear l_1/l_∞ norm penalties. The controller allows tuning performance and the constraint violation degree to guarantee stability. Moreover, feasibility can be ensured in a large region of the state space based on imposing hard constraints in the design. Zeilinger et al [173] also proposed a framework which combines the proposed soft constrained formulation with a robust approach, to satisfy constraints in the presence of a disturbance, while using soft constraints to ensure feasibility and stability for exceeding disturbances. The authors showed that the stability results extend to the case of a significant size of disturbances to ensure safety and feasibility during on-line operation.

Combined linear and quadratic penalties methods

De Oliveira and Biegler [35] discussed three different relaxation strategies of output constraints using penalty functions. The first is the use of the quadratic penalty function, in which a scalar penalty parameter and a squared slack variable which is a measure of the original constraint violation are included in the cost function. The idea is to determine an appropriate finite maximum value of the penalty parameter that can be used for tuning and tolerated for stability. De Oliveira and Biegler [35] showed that the stability of the soft constraint MPC problem is equivalent to that of the unconstrained MPC problem, with a simple change in the tuning parameters. However, the drawback of this approach is that it is combinatorial in the length of the predictive horizon due to the dependence of the penalty parameter on the current active set.

The second approach is the use of the exact penalty function. This requires the penalty parameter to be sufficiently large to satisfy the exact function. This can be done by ensuring that a finite value of the penalty parameter must be larger than the norm of the Lagrange multipliers of the constraints in the original problem [43]. This can lead to a constrained formulation which has the same bounded stability properties as exist in the absence of the soft constraints. De Oliveira and Biegler [35] showed that this exact penalty approach is simpler to apply to time-varying and non-linear systems compared with the quadratic penalty approach. However, it is difficult to decide exactly how large the penalty parameters should be, since the Lagrange multipliers depend on the current state (Maciejowski [90]).

The third approach is to use the l_∞ norm. In this case, the result obtained from the previous approach is extended to the case of this approach and it was shown that a stronger bound for the Lagrange multipliers exist. De Oliveira and Biegler [35] reported that the exact penalty strategy using either the l_1 norm or the l_∞ norm can avoid unnecessary constraint violations and that these are well-suited for soft constraints.

Scokaert and Rawlings [152] presented an approach for handling soft state constraints for addressing infeasibility, by incorporating a combination of l_1 norms and squared l_2 norms of the state constraint violations into the MPC cost function. In order to ensure the exact penalty property [43], Scokaert and Rawlings [152] introduced a method that determines a conservative state dependent upper bound for the optimal multiplier for the original MPC problem by using the Lipschitz continuity of the quadratic problem [59]. However, it is unclear exactly how to implement this for a whole feasible state space [69].

Scokaert and Rawlings [154] analysed the soft constraint approach introduced by [175] and explored its strengths and weaknesses. The authors showed that it is computationally cheap, requiring a simple QP and stabilizing , with the Lipschitz continuous in the state. However, they also showed that in some cases, such as minimum phase plants, the optimisation of constraint violations is a multi-objective problem since, as the size of the violation is reduced an increase in duration occurs. Moreover, it provides a significant mismatch between open-loop predictions and closed-loop behavior as well as tuning difficulties when choosing the weight matrices.

In order to overcome the difficulties encountered in [175], Scokaert and Rawlings [154] proposed a new soft constraint MPC approach, where the total sum of the square values of the constraint violations rather than the peak value violation is penalized at each time step along the prediction horizon. This is achieved by adding a combined quadratic and linear term of constraint violation penalization to the cost function. The purpose in penalizing the weighted l_1 norm of the sum of the predicted constraint violations to allow the use of exact penalties, while adding the quadratic penalty to enhance flexibility. The quadratic penalty also results in a well-posed quadratic program. Scokaert and Rawlings [154] showed that it is possible to achieve trade-off between the peak value and duration of the violations by tuning the quadratic soft constraint penalty and that open-loop predictions match the nominal closed-loop behavior for all choices of tuning parameter.

Alternative methods

A novel soft output constraints approach was used in [116] to develop a robust constraint MPC controller for linear systems that provides robustness against model plant uncertainty. In this work, output constraints are relaxed by introducing a penalty function such that it is zero or almost zero within the dead zone within the soft limits. This dead zone reduces the sensitivity of the controller to noise and uncertainty when the process output approaches its target. Prasath et al [116] demonstrated that the proposed controller does not degrade much in the nominal case, but improves significantly in the case of plant model mismatch. They also demonstrated that the controller can be efficiently implemented into the process with the significant model uncertainties that arise from the plant such as a Cement Mill process [117].

Recently, an alternative technique of output soft constraint for fast algorithms was proposed by [135]. The idea is to introduce a method which completely avoids the inclusion of any additional slack variables. The method is based on the introduction of an approximated violation penalty directly into the cost function, using the Kreisselmeier-Steinhauser function [76]. Richards [135] demonstrated that the proposed formulation can be applied to the fast algorithm of [134] efficiently, retaining the favourable matrix structure of the original fast

MPC. However, the proposed method comes at the cost of the approximation of the soft constraint penalty [45].

Summary: All of the soft constraints algorithms described in this subsection, except for the algorithm of [135], share the common concept of constraint relaxation, that is, the inclusion of a slack variable within the cost function and the penalisation of output or state constraints but they differ regarding the type of the penalisation. It has been shown [35] that the linear penalty (l_1 norm or l_∞ norm) is well-suited for constraint relaxation. However, all of these methods focussed on constraint management rather than the cause, i.e an over-ambitious target.

3.3.3 Set point management and reference governor methods

Another popular way of addressing feasibility and stability problems during rapid set point changes is the use of the set point management and reference governor methods [145]. In this manner, several results have been obtained for the feasibility and stability of MPC for tracking scenarios [143, 10, 158, 126], which are described as follows.

In [143], a Constrained Stable Generalised Predictive controller (CSGPC) for SISO plants is presented; the proposed controller ensures feasibility by deploying, temporarily, an artificial reference as a degree of freedom (d.o.f) and convergence is ensured by means of a conservative constraint based on the artificial reference. This approach is extended in [137] for more general terminal conditions and to make use of invariant sets to handle the constraints. In this work, it is demonstrated that changes to the loop target can provide a highly effective mechanism for increasing the volume of feasible regions; consequently an artificial target can provide a more useful degree of freedom within MPC than the more normal choice of future control increments.

An alternative, well-known approach for dealing with temporary infeasibility due to changes in the target is a command governor (CG), whose action is based on the current state, set point and prescribed constraints [10, 50]. One example of this approach is the addition to the system of a non-linear low-pass filter, which is selected to ensure the satisfaction of the constraints while retaining offset-free tracking behaviour. The command governor approaches have a strong synergy with the closed-loop paradigm implementations of MPC. Their main weakness is their simplicity; that is, they sacrifice transient performance to ensure an effective but simple approach.

Similar ideas have been applied in conjunction with non-square techniques [158] which investigate the impact of non-square systems on feasibility in tracking problems in linear MPC. In this work consideration was given to cases where the target was unreachable in the steady-state. This line of approach was extended in [157] to consider the impact of uncertainty on the control law's ability to find the optimum steady-state for unreachable targets.

In [25], a novel algorithm for tracking a piecewise constant reference for a linear system subject to input/state constraints and bounded disturbance was proposed. The core concept is to deploy predictive regulator whenever feasibility holds and deploy feasibility recovery in the case of feasibility loss due to set point changes. Chisci et al [25] demonstrated that the proposed algorithm can provide good tracking performance with robust constraint satisfaction and yield a reasonable on-line computational burden.

In [3], a novel approach for a discrete time linear system under input and state constraints to track a piecewise reference signal was introduced. The idea is to combine the concept of a command governor with the conventional MPC algorithm. This approach is effective but only for specific scenarios where the set point is fixed.

In [86, 40], a slight variation on MPC for tracking changing constant references for both constrained linear and non-linear systems was presented. These controllers ensure feasibility by means of adding an artificial steady-state and input as a degree of freedom of the optimization problem. Convergence to an admissible target steady-state is ensured by using a modified cost function and a stabilizing extended terminal constraint. Optimality is ensured by means of an offset cost function which penalizes the difference between the artificial reference and the real one.

A subtlety of interest here is the recovery of local optimality [40] despite the use of a bias term in the performance index. It is proved that the proposed controller steers the system to the target, if this is admissible. If not, the controller converges to an admissible steady-state optimum according to the modified performance index. This approach was extended in [41] by considering an alternative offset cost function based on the infinity norm, as it was shown that this enabled the recovery of a local optimality property while retaining all of the key properties of the original formulation.

In [125], a new infinite horizon MPC formulation for the case of active steady-state constraints is implemented and discussed. This new formulation is based on an iterative algorithm that determines the optimal solution to the control problem within a user-specified tolerance.

In [126], the case of unreachable set points is considered; here, the authors propose that the performance index should be based on the distance from the unreachable set point rather than the artificial, reachable one. The authors proved the asymptotic stability and the convergence to the steady-state values with a desired tracking response, despite the fact that the performance index is technically unbounded.

A novel predictive control formulation for tracking a known periodic reference signals was proposed in [84]. The concept is to consider both the predicted system input and the pseudo-reachable target as the degrees of freedom and, moreover, to consider a performance index that includes a penalization of both the tracking error with the pseudo reference and the difference between the pseudo reference and the known future reference. Stability is guaranteed by adding a terminal constraint to the predicted trajectory alongside the constraint that forces the artificial reference to be periodic. Limon et al [84] proved that the proposed controller provides a recursive feasibility with the controlled system that ensures the hard constraints satisfaction. It is also proved that the proposed controller is asymptotically stable and converges to the most reachable target trajectory.

A simple model predictive control framework for a random reference tracking linear systems was presented in [39]. The key idea is to add a constraint to the optimisation problem and to select a virtual reference that maintains the trajectory in a specific set. The proposed approach with the virtual reference ensures that all trajectories remain within a compact set.

A different strategy when tracking a random reference was considered in [93]. The proposed control strategy also employs an artificial reference as a state variable rather than a control variable. The controlled system is subject to input and state constraints as well as the following terminal constraints. In this work, [93] showed that the state converges to the steady-state value and that the tracking error is bounded.

In [87], a new MPC method for tracking arbitrary periodic references was presented. The concept is to combine the trajectory planner with the tracking MPC formulation in a single optimization problem such that the degrees of freedom are a planned reachable trajectory and the input signals sequence is as predicted. The key point here is that the loss of feasibility is avoided since the set of constraints of this optimization problem do not rely on the reference signal. By using a slightly modified Lyapunov theorem [127], Limon et al [87] proved that the controller can guarantee asymptotic stability and ensure feasibility even in the presence of sudden target reference changes.

Recently, the stability and recursive feasibility of nominal MPC under reasonable conditions were shown in [95], and [56] showed that recursive feasibility and stability can be ensured avoiding a terminal stability constraint and that many properties of terminal constrained NMPC schemes can also be derived without terminal constraints. Moreover, they demonstrated that, by avoiding terminal constraints, NMPC can yield controllers with large operating regions even for very short optimization horizons. On the other hand, [93] argued that the inclusion of explicit or implicit terminal constraints in the optimal control problem is essential for ensuring both stability and recursive feasibility.

More recently, Santos [151] introduced a modification to the MPC formulation for tracking piecewise constant references using a simplified parametrisation of steady-state, avoiding the use of additional decision variables in the optimisation problem by building on [86, 40]. In this work, the artificial references are obtained from the terminal state and the terminal control by means of the simplified parametrisation. Moreover, the author also proposed a modified algorithm that ia able to recover the feasible region provided by the MPC for tracking with an artificial target [86, 40]

Summary: The set point management algorithms typically deploy two different concepts: one is the use of the reference governor method while the other is based on the inclusion of an artificial target in the cost function for the associated optimisation problem. The weakness of command governor approaches is their simplicity; that is, they sacrifice transient performance to ensure an effective but simple approach. On the other hand, artificial reference approaches possess strengths in that they can be deployed for both reachable and unreachable targets for time-varying systems.

3.3.4 Section summary

We can now summarise the MPC algorithms discussed in Subsections (3.3.2, 3.3.3) that guarantee feasibility and the stability of the closed-loop of the associated control as follows.

Infeasibility and instability problems which are encountered due to a rapid set point changes and disturbances or modelling error can be solved efficiently by using soft constraints or set point modification methods. While the work mentioned above achieved good results, ensuring feasibility retention and stability guarantees, nevertheless, the range of solutions and approaches presented in the literature for handling future target information remains limited. Specifically, there has been a tacit avoidance of the impact of feed-forward, as the solutions have generally focused on cases where the future target is assumed to be constant, which implicitly means that no feed-forward information about future target changes exists. This thesis will focus on the concept of artificial targets since this concept focuses on the over-ambitious targets. This will be discussed in Chapter 5.

3.4 Offset-free tracking model predictive control

One of the main issues that must be considered when developing of efficient industrial MPC control approaches is the offset-free tracking problem, which can be defined as the problem of tracking a reference set point and achieving zero offset in the presence of persistent disturbances and plant model mismatch [100, 110, 91]. Several strategies have been proposed to solve this problem. This section will focus on reviewing the developed algorithms within the tracking scenarios.

3.4.1 Disturbance model

In order to account for the disturbances or plant model error, a standard strategy is to augment the model of state with the model of disturbance using an estimator. Several algorithms have been proposed based on this strategy, which can be described as follows.

Muske and Badgwell [100] presented a solution to the problem of offset-free control by adding a general disturbance model for linear MPC, which contains unmeasured disturbances entering through the process input, state, or output. The authors showed that the states of the process and unmeasured disturbance model must be estimated simultaneously by using an appropriate stable estimator. They also showed that the steady-state input and state target vectors, which are required for removing the effect of the estimated disturbances, can be determined by using a QP problem solution. For the combination of the estimator, steady-state target calculation and controller dynamics, Muske and Badgwell [100] derived sufficient conditions to achieve zero offset control, such that the augmented model must satisfy the detectability condition and the total number of unmeasured disturbances and the number of measured outputs must be equal. It is demonstrated in this work that offset-free control occurs in the presence of significant model error and suggested that control performance may be improved by using state or input disturbance models.

In [110], an MPC algorithm which solves the offset-free control problem was designed by augmenting the process model by integrating disturbances and using a steady-state Kalman filter to estimate both the state and the disturbance based on the availability of the measurements of the plant. In this work, general conditions which allow the design of an offset-free algorithm were derived such that the augmented system must be detectable and the maximum number of the added disturbances must be equal to the number of measurements. Pannocchia and Rawlings [110] demonstrated that all detectable disturbance models guarantee offset-free performance. The results reveal that the plant follows the non-linear model as well as the linearised model used by the controller. However, it was shown that, for ill-conditioned processes, the request for a zero steady-state offset in all output variables can lead to closed-loop instability in the case of incorrect model identification. In this manner, Pannocchia and Rawlings [110] suggested that in order to maintain closed-loop stability, it is possible to relax the offset-free performance related to the least important variables .

Pannocchia and Kerrigan [108] presented a new MPC algorithm for controlling constrained linear systems with time-varying targets in the presence of unmeasured, bounded disturbances. This algorithm is based on designing a dynamic controller, which includes a dead-beat observer for disturbance estimate, target calculation for steady-state values and a static state feedback gain to regulate the plant model to a steady-state. The key idea in this work is to incorporate both the transient and limiting effect of all allowable future disturbance and set point sequences to the controller formulation. Pannocchia and Kerrigan [108] demonstrated that the proposed algorithm is guaranteed to achieve a zero steady-state offset as the disturbances reach an unknown steady-state value even if modelling errors exist. They also showed that offset-free control holds independently of the actual plant dynamics.

A novel method for addressing the offset-free model predictive control problem was proposed in [107]. The method is based on the integrating process model design with its associated observer. In order to minimize the impact of the unmeasured disturbances and plant model error on the output prediction, the author introduced a dynamic observer nominal design by means of an H_∞ control problem solution. In order to create a compromise between the effectiveness in achieving zero offset control and low noise sensitivity, a scalar tuning parameter is performed in the design. Pannocchia and Bemporad [107] showed that when offset-free control is required, the designed observer is equivalent to selecting an integrating disturbance model and a state observer for the augmented system.

In [2], a new MPC offset-free tracking algorithm design was presented. This algorithm is able to enforce output offset-free tracking of the reference input with constraints satisfaction when unmeasured disturbances are available. This algorithm consists of two parts. In the first part, a stabilizing linear time invariant controller is designed to achieve a zero offset tracking advantage due to the dynamics structure augmentation. In the second part, this advantage is then supported by selecting the MPC controller design in which the state and input constraints are explicitly included in the MPC controller and therefore robust constraint satisfaction is guaranteed. In addition, the presented algorithm can satisfy the offset-free tracking of all reference inputs, which can be described in the rational transfer function form with no disturbance estimating requirement.

In [91], the offset-free MPC problem for tracking a fixed reference was addressed. In this work, an additional disturbance was added to the system to be estimated using a linear disturbance observer and simple conditions that ensure that the offset-free performance of both the estimator and the controller has been derived. Maeder et al [91] discussed the case whereby the number of disturbances and the number of measured variables are equivalent, and reported that this may lead to greater complexity when more disturbances than the number of controlled variable are introduced. Moreover, they discussed the case whereby the number of disturbances is equal to the number of tracked variables but smaller than the measured variables and designed an observer such that the offset is eliminated in selected variables, thereby providing an MPC with minimal complexity. Finally, Maeder et al [91] provided insights into the zero steady-state offset when the performance index is one or infinity norm objective functions, and when explicit MPC is implemented for complexity reduction.

It has been shown in [122] that, for linear models, the choice of the disturbance model has no effect on the closed-loop performance if suitable covariances are used to specify the state estimator. In this manner, Rajamani et al [121] presented the Auto-covariance Least Square (ALS) technique [121] to calculate the appropriate covariance of the disturbance model based on steady-state data irrespective of the actual unknown plant disturbance source. The estimated covariances are then used to calculate the estimator gain. It is shown that the incorrect selection of the disturbance source in the disturbance model is compensated for by the estimated covariances and the resulting gain of the estimator. It was demonstrated that as the estimator gain is determined from the covariances estimated using the auto-covariance of the data, the inputs equal the outputs and hence are indistinguishable by means of two different (either deterministic or stochastic) disturbance models.

Maeder and Morari [92] presented a method for offset-free model predictive control for reference tracking and disturbance rejection. They considered reference and disturbance signals generated by arbitrary, unstable linear model dynamics. Maeder and Morari [92] constructed a disturbance model, which includes the reference dynamics and input disturbance and proposed an algorithm that computes the target trajectory. Under this method, a linear observer was employed to estimate the state and disturbance, assuming that the augmented system is observable. The key points here are that the disturbance model must satisfy the internal model condition (that is, the dynamic matrix of the disturbance model must contain the dynamic matrix of the reference model) and the augmented system must be observable. Moreover, the target trajectory conditions must be derived to be added to the MPC problem such that the modal composition of the reference and disturbance is trivial. Offset-free is achieved assuming that the disturbance and reference dynamics are appropriately embedded into the prediction model , thereby providing reference feasibility.

A novel MPC offset-free formulation was developed in [62]. The core idea is to combine one basic MPC controller with another MPC controller with integral action, to achieve zero offset control and also deal with known and unknown disturbances. One controller is appropriate for control in the presence of known disturbance while the other can reject unknown disturbances. In order to maintain the task of the proposed controllers, the balance between the two actions is purchased by combining the two performance indices. Horvath et al. [62] applied the proposed controller structure to control the level of open channels and showed that the developed offset-free method is able to eliminate the steady-state offset, while also taking into account the known and unknown disturbances.

3.4.2 State disturbance observer

An alternative strategy for the offset-free problem is the state disturbance observer. This strategy is not based on an augmented system in which the current state is estimated using an observer with an additional output correction term, which is necessary to ensure offset-free tracking. Several algorithms based on this strategy may be found in the literature to date, so we will describe more recent approaches in the following.

An offset-free MPC approach with linear state space system was presented in [161] for constant unmeasured external and plant model mismatch and asymptotically constant references. This approach is based on the technique of state disturbance prediction; that is, the use of a suitable state disturbances structure in the model for output state prediction.

Tatjewski [161] recalled that in the case of a measured state, using constant state disturbance prediction enables offset-free control without the need for a disturbance observer or filter. Regarding the unmeasured state (where the process outputs alone are measured), only the process state must be estimated, using a general Kalman filter with the addition of a suitable correction term for the output prediction. The author proved that the use of this technique, in this case, enables offset-free tracking.

Recently, Tatjewski [162] extended the work of [161] to non-linear state space systems. The presented approach uses a formulation for non-linear models and for the MPC cost function, which includes predicted control errors and process input increments. This work provided a theoretical analysis of the offset-free tracking property with less restrictive conditions than is the case under the conventional approach. The advantage of the proposed controller is that it is simple and easy to design with less restrictive applicability conditions than the conventional approach with extended process and disturbance state estimation. Another advantage is the avoidance of the need to select a limited number of disturbances and their inclusion in the process model.

3.4.3 Velocity form

Another alternative strategy for offset-free tracking problem is to compose the enlarged state by the state increment and the output error, while the manipulated variable is the control increment. In the literature, this strategy is called the velocity form model. Various solutions based on this strategy have been proposed in the literature. An overview of some of these solutions is presented below.

The velocity algorithm was studied under the general LQ regulators formulation in [109]. The idea is to derive the initial estimate and its covariance matrix in the velocity form based on the initial estimates in the state space traditional and use the augmented variables in the LQ regulator formulation to define a suitable control law. It was shown that offset-free control properties are guaranteed without the use of an explicit disturbance model, since the target of the state increment is always zero even in the presence of plant model mismatch [111]. Pannoccia and Rawlings [109] compared the results obtained by using the velocity algorithm with the standard state space model and showed that, when the velocity algorithm is used with correct initialisation, the disturbance entering the plant is completely rejected while, when using the traditional state space model, the offset occurs.

In [53], a discussion about ensuring offset-free control for a linear stable state space model, using the velocity form model was presented. The author showed that the system described by the complete velocity form model, where the states and input increments are considered, can lead to offset-free control under two conditions. One condition is to obtain unbiased steady-state predictions, which is achieved by adding an integral action to the observer. In this manner, the author demonstrated that the integrating model embedded by the complete velocity model into the observer ensures that the output estimation reaches the true plant output value. The other condition is to design a well-constructed MPC optimisation problem. It is shown in this work that the cost function contains the augmented state vector, which is penalised to achieve zero state increment and the output is equal to the set point.

In [16], a new model predictive control algorithm for offset-free tracking piecewise constant reference signals in the presence of unknown disturbance and model plant mismatch was proposed. In this algorithm, the offset-free tracking is guaranteed by enlarging the control loop with integrators and describing the system to be controlled in velocity form. Gonzalez et al [53] showed that , under the proposed algorithm, the controller can track any admissible reference signal without any error and reject the influences of both plant model mismatch and the unknown disturbance. They also showed that stability is guaranteed through a suitable definition of the auxiliary control law and the terminal set used in the MPC problem framework.

Recently, Pannocchia [106] reviewed the disturbance model approach and the other two alternatives, the state disturbance observer model approach [161] and the so-called velocity form model approach, within a coherent framework. It was shown in this work that the two alternative formulations are considered particular cases of the disturbance model approach. By comparing the two alternatives with the disturbance model method individually, Pannocchia [106] has shown that by using suitable matrices for disturbance model matrices and for the prediction output and augmented observer gain vectors, both the augmented systems and the augmented observer dynamics satisfy the detectability and the dead-beat conditions respectively, and are equivalent to the alternatives. They have also shown that the simple tuning for the state disturbance model approach can be applied to the three approaches.

3.4.4 Section summary

Now, we will summarise some of the strengths and weaknesses of the offset-free control methods described in Subsections (3.4.1-3.4.3).

The offset-free strategies described above share the concept of including an integral action in the control loop but each one has its own specific strengths and weaknesses. The disturbance model method is simple and straightforward to generalise but its weakness is that it is unclear how it is effectively chosen in suitable MPC algorithms. In addition, the observer may encounter tuning-related difficulties [161].

On the other hand the velocity form method is rarely used in offset-free control since, in principle, a problem may arise when using the infinite prediction horizon to ensure that the loop is stable. This occurs when the integrating states which include the velocity model result in a permanent output error, which is hence dramatically doubled in regard to unbounded costs [109]. Moreover, using this form can lead to computational costs due to the increase in the state dimension. Nevertheless, it offers an advantage in that it does not require either the choice of a disturbance model or a steady-state calculation. However, recently, it has been shown [106] that it can be treated as a special case of the disturbance model. This would avoid the question of which is more effective or beneficial. In this thesis, we will deploy the disturbance model method since it seems to be popular, effective and consistent with the assumption that disturbance varies slowly.

3.5 Preview (advance knowledge) and feed-forward

In order to review the inclusion of feed-forward in the model predictive control, we define three important components [54] related to this; namely, preview, which is advance information about future target or disturbance changes. It is clear that such information is useful for feed-forward control design since it allows the controller to prepare for target or reference changes. It is shown in [98] that preview offers significant performance advantages; feed-forward, which is part of the control action applied to the reference signal without correction for the measured response; and feedback, which is the closed-loop control action with error correction for the observed response. In this thesis, we use the terminology, advance knowledge or preview.

MPC has the potential to include feed-forward information systematically rather than as a separate design. However, assuming, for simplicity that the feed-forward (FF) is related to tracking scenarios (FF for disturbances would give equivalent observations), the literature as a whole is relatively vague on how to make effective use of information on future target values within MPC optimisation.

The relative neglect of feed-forward arises partly because the most common assumption adopted is that, for any given sample, the future target is assumed to be constant. Typically the literature focuses on scenarios with constant targets but as a consequence aspects of some results may not be representative when the target changes frequently. Moreover, even where target changes are considered, most of the literature ignores the feed-forward term since it tacitly assumes that there is no advance information on any changes that do occur. Nevertheless, a few works have explicitly considered the impact of including advance information on target values into an MPC optimisation and this forms the focus of the current thesis.

Early work [139] established that the default feed-forward from a GPC algorithm could be a poor choice, but may be improved by a different parametrisation of the degrees of freedom or reducing the amount of advance information used in the optimisation. The focus was solely on step changes in the target.

These insights were extended slightly in [144, 166] where it was demonstrated that for a finite horizon algorithm with a small control input horizon n_u, the best feed-forward depends upon the shape of the set point trajectory and, more specifically, is not usually the same as the default that arises from an MPC optimisation. Finite horizon algorithms remain important due to their prevalence in the industry but obviously with the increasing trend towards dual-mode algorithms, these must also be considered and here it was noted that the best choice of feed-forward was less dependent on the target trajectory.

In [54], a novel strategy for MPC design which incorporates feedback, reference feed-forward and preview was formulated. The concept here is similar to that in [144, 166], as suggested in [54], which separated the feed-forward design from the feedback loop controller. In this strategy, the feed-forward was selected as a signal rather than a compensator to optimise performance/tracking whereas the loop controller focused on robustness. Goodwin et al [54] demonstrated that for certain scenarios proposed controller vastly improves the performance of time-varying reference tracking and that the model uncertainty is corrected and the majority of the tracking performance recovered when the constraints are included in the feed-forward optimisation stage. Nevertheless, this work did not discuss how much advance information about the target was useful and instead focused on the concept of there being a two stage design. Moreover, the feed-forward signal was designed as an on-off and used in its entirety with the feedback added on top to correct for uncertainty, whereas in fact the feed-forward contribution could be modified as new information on the target becomes available.

In [21], the concept of preview and feed-forward in MPC, introduced in [54], was extended to a framework that guarantees robust stability. In this work, the author considered constant reference signals to be tracked. Carrasco and Goodwin [21] argued that the proposed strategy is always useful or at least not detrimental, since feedback optimisation can correct the additional feed-forward signal at any time if the system's stability is endangered by this feed-forward signal.

Summary: The importance of incorporating a feed-forward compensator with preview, particularly for constrained MPC, has been described above as providing good results. However, there remain some outstanding questions. The extent to which one can make statements about how much advance information on the target is useful remains unclear, which is one of the focuses of the current thesis. Following this, little consideration has been made in the literature outside of [54] regarding how off-line decisions on the handling of feed-forward information can be effectively incorporated into MPC optimisation. This will be discussed in the following chapter.

3.6 Overview of robust MPC

Robust model predictive control has often been studied as a regulation problem [13, 71, 34]. Such approaches are based on Min-Max optimisations using linear matrix inequalities techniques. Robust stability properties are studied in [23].

The use of LMI optimisations implies a substantial on-line computational load and hence, in [75], an alternative algorithm was proposed which provides a significant reduction in computational costs by adding degrees of freedom and utilising off-line computations effectively. Nevertheless, the implicit reliance on ellipsoidal invariant sets restricts the regions of applicability of these approaches.

In [114, 113], a mechanism was proposed for developing robust invariant sets based on linear rather than quadratic inequalities. This simple set definition allowed a volumetric increase of the infeasible regions and for the definition of a simple robust MPC algorithm for LPV systems that requires only quadratic programming optimisation while retaining the core properties of recursive feasibility and guaranteed convergence.

Another thread in robust predictive control research is the so-called tube based approaches [80, 123, 24]. The idea is to form a tube (equivalently a set) which contains all of the possible evolutions of the system predictions for a given level of uncertainty. However, such approaches have focused primarily on uncertainty due to exogenous signals such as disturbances and also require significant off-line computations which are challenging with large state dimensions and large disturbances and thus are considered no further here.

In [94], a simple output feedback model predictive controller was developed for constrained linear systems with input and output disturbances, using a Luenberger observer as the state estimator. The proposal used an invariant set that bounds the estimation error but notably relied on a tube-based robust model predictive controller approach.

A critical observation is that the vast majority of work on robust MPC has focused on the regulation scenario that has a fixed set point/target. However, in many scenarios, a key role of MPC is to help to track the target changes in the presence of input and state constraints. One key concept here is *advance knowledge* [139]; that is, how many samples into the future is the set point trajectory known and available to the MPC algorithm. Therefore, this thesis will focus on the potential robust MPC approaches which explicitly include the tracking scenario. A key challenge when the set point changes is to ensure both recursive feasibility and stability in the presence of model uncertainty.

Consideration was given in [158] to cases where the target was unreachable in the steady-state and also to the inclusion of non-square linear systems. Thus a key objective was to consider what modifications to an MPC algorithm are needed to guarantee feasibility during target changes. The chosen line of approach was extended in [156] to consider the impact of uncertainty on the control law's ability to find the *optimum* steady-state for unreachable targets. However, these works considered fixed targets only, in the absence of any advance knowledge.

In [5], a novel formulation of a robust output feedback model predictive controller was proposed to track piecewise constant references for a linear system with additive bounded uncertainties on the states. The proposed MPC can steer the uncertain system in an admissible evolution to any admissible steady-state; that is, under any change of the set point. This controller is applied successfully in industry [6], but again assumes no advance knowledge.

The design procedure for a robust MPC for tracking constrained linear systems with additive disturbances is presented in [85]. The paper utilises the notion of a tube of trajectories, to achieve robust stability and convergence and also uses LMI based design procedures and algorithms for the calculation of invariant sets. Again, no advance knowledge is considered.

A robust model predictive control algorithm for solving the tracking and infeasible reference problems for constrained systems subject to bounded disturbances is presented in [17]. A robust tracking MPC for input constrained uncertain systems, that is, based on the construction of a feasible and invariant set is introduced in [105, 83].

Summary: Different robust MPC approaches have been described in this section. While these approaches give a good closed performance and are guaranteed to satisfy the process constraints, no advance knowledge is considered. In this thesis, we propose a modified robust MPC formulation that can handle parametric uncertainty considering future information about target changes.

3.7 Literature review on improving mp-QP

It is shown in the literature that the linear MPC optimization problem, with a quadratic cost function and linear state and input constraints, can be solved via multi-parametric programming (mp-QP) techniques for instance; [14], [149]. It is shown in this literature that the obtained controller is piecewise linear and continuous and provides all of the stability and performance properties of MPC. Furthermore, the on-line computation is reduced to a simple linear function evaluation rather than the expensive quadratic program. The benefit of this solution is the potential reduction of the computational burden that arises from the on-line optimization.

In [66], an explicit solution to the infinite-horizon LQR problem with state and input constraints based on receding horizon real-time quadratic programming was developed, eliminating the need for real-time optimization. The computer memory and processing capacity requirements of the explicit solution are addressed by suggesting a strategy, based on a suboptimal choice of a finite horizon and imposing additional limitations on the allowed switching between active constraint sets on the horizon. It is shown that the resulting feedback controller is piecewise linear, in order to analyse stability and performance as well as efficient real time implementation.

In [11], an explicit RHC law for linear systems with input and state constraints was computed by using the mp-QP technique. In this work, the problem of finding approximate solutions to mp-QP was addressed, where the degree of approximation is arbitrary and allows a trade-off between optimality and a smaller number of cells in the piecewise affine solution. Furthermore, the obtained law provides closed-loop stability and constraint fulfilment.

The properties of the polyhedral partition of the state space induced by the multi-parametric piecewise affine solution were discussed in [163] using a new mp-QP solver. Compared to the existing algorithms, this approach adopts a different exploration strategy for subdividing the parameter space, avoiding unnecessary partitioning and QP problem solving, with a significant improvement in efficiency.

An efficient algorithm for mp-QP was presented in [9], using, Karush Kuhn Tucker (KKT) conditions to characterize polyhedral critical regions and the corresponding optimal solution. However, in this work the unnecessary partitioning of the parameter space was avoided by using a direct exploration strategy. The neighbourhood of the initial critical region was explored by crossing each of the faces and checking if a feasible neighbouring critical region exists. This procedure is then repeated iteratively with all newly generated regions.

Another approach was introduced in [79] to reduce the mp-QP complexity. The key idea is to separate the regions that contain a saturated control law. This can be performed by using the linear programming (LP) method. However, the drawback is that these approaches are applicable only for small systems.

In [57], a new approach was presented for reducing the combinatorial complexity of the optimization method that resulted in the previous approaches. The main idea is to exploit all of the space using a minimum number of space partitions.

A novel algorithm for the mp-QP optimization problem was presented in [148]; this algorithm enumerates all of the possible combinations of the active sets. Although the algorithm is systematic and effective, it may result in a combinatorial complexity difficulty.

In [70], an alternative approach was proposed to minimize the combinatorial complexity.

This approach implements an algorithm based on the relaxation of infeasible constraint combination and exploiting the geometric properties of the mp-QP constraints to increase the efficiency of the combinatorial enumeration. However, this algorithm may cause an increase in the complexity of the constraints polyhedrons.

Summary: Critically, the existing work has not considered how to handle systems with lots of states and information about the future target, so one can propose a new formulation that addresses this challenge. This will be discussed later in this thesis.

3.8 The key observations on the literature review

It was demonstrated that, while many solutions have appeared in the literature, these do not offer comprehensive or systematic solutions to several key scenarios. Consequently, this section presents several issues for further study, as detailed below:

1. Infinite horizon or OMPC tracking algorithms are effective but in general have been limited to fixed set points scenarios.

2. In tracking OMPC algorithms the effective use of future knowledge about set point changes is ignored in the literature for both nominal and robust cases. In Chapter 4, we will introduce an efficient algorithm that makes effective use of future set point information for nominal cases. This efficient algorithm is extended to be implemented in robust cases, as will be discussed in Chapter 6.

3. The theory of the maximal admissible sets (MAS) for infinite horizon algorithms for constrained systems with time-varying targets and uncertain systems has not been fully exploited. We will illustrate how to construct an admissible set for systems with time-varying targets for both nominal and uncertain cases in Chapters 4 and 6.

4. The issues of feasibility and stability are discussed in the literature. However, the scenario where the target is unreachable is not widely considered in the literature. Therefore, we will study in Chapter 5 the feasibility and stability of both reachable and unreachable targets for the nominal case, using the artificial target approach.

5. Modifications to an MPC algorithm are needed to guarantee feasibility during target changes for the robust case. The design of an efficient algorithm will be explored in Chapter 6.

6. It is possible to implement the parametric (mp-QP) technique for the implementation of feed-forward with advance knowledge, using an appropriate algorithm but this has not yet been discussed in the literature. This algorithm will be introduced and illustrated in Chapter 7.

7. Feasibility issues with the inclusion of the current input and steady state offset within a parametric approach will also be investigated in Chapter 7.

Chapter 4

FIXED FEED-FORWARD DESIGN WITHIN DUAL-MODE APPROACH

This chapter constitutes a core contribution to the thesis. It was mentioned in the previous chapter, Subsection 3.1.3, that the default feed-forward arising from a conventional MPC algorithm may be inadequate because the assumptions implicit in the optimisation are relatively limited and may only be valid for fixed set points. If the future set point changes, then the optimization and degrees of freedom within it need essential modification.

It is logical to consider whether a two stage design would be a better choice for feed-forward, in other words: (i) first design the feedback loop for robust performance and (ii) second, design a feed-forward to give optimum tracking, assuming that the inner loop is known. Regarding this concept, it is necessary to clarify how exactly the design can be created and to what extent such a design can handle the constraints. Consequently, a key objective is to define an algorithm that can make effective use of a predefined feed-forward; that is, to embed this into the on-line optimisation and evaluate this approach compared to more conventional methods.

This chapter is organised as follows. Section 4.1 illustrates how to include time-varying target information into dual-mode algorithms. Section 4.2 presents the derivation of a control law for an OMPC approach, using future information about target changes. Section 4.3 introduces two useful proposals for the effective use of advance knowledge in feed-forward design. Section 4.4 illustrates numerically the use of advance knowledge on the OMPC for an unconstrained case. Section 4.5 presents a novel formulation which handles the constraints for feed-forward with advance knowledge. Section 4.6 presents numerical examples for embedding advance knowledge with constrained OMPC. Section 4.7 discusses the insight of embedding advance knowledge for both constrained and unconstrained cases. Finally, Section 4.8 presents the conclusions to the chapter.

4.1 OMPC dual-mode with a time-varying target

The literature on dual-mode MPC tends to assume that the expected steady-states x_{ss}, u_{ss} of (2.32) are fixed which effectively means that the target is fixed and no advance information is introduced into this performance index. This section explores how to handle time-varying targets in feed-forward design within an OMPC dual-mode formulation.

This section shows how to define an autonomous model for model predictions, a performance index and a control law for an OMPC dual-mode approach, including future information about target changes. It will also show explicitly how the incorporation of future information about target changes has an impact on the optimum feedback.

4.1.1 Closed-loop predictions for a time-varying target

In Chapter 2, state and input predictions were described in terms of the deviation variables (2.59). However, these predictions are only suitable for a fixed target. In this chapter, we aim to describe predictions which incorporate time-varying targets. In this case, one can rewrite equation (2.59), such that the target is no longer fixed as:

The state predictions are given by:

$$\begin{cases} x_{k+i+1} - x_{ss|k+i+1} = A(x_{k+i} - x_{ss|k+i}) + B(u_{k+i} - u_{ss|k+i}), & \text{........} \quad i \le n_c \\ x_{k+i+1} - x_{ss|k+i+1} = A(x_{k+i} - x_{ss|k+i}) + B(u_{k+i} - u_{ss|k+i}), & \text{..........} \quad i > n_c \end{cases} \tag{4.1}$$

Similarly, the input predictions can also be described as:

$$\begin{cases} u_{k+i} - u_{ss|k+i} = -K(x_{k+i} - x_{ss|k+i}) + c_{k+i}, & \text{............} \quad i \le n_c \\ u_{k+i} - u_{ss|k+i} = -K(x_{k+i} - x_{ss|k+i}), & \text{..........} \quad i > n_c \end{cases} \tag{4.2}$$

where $u_{ss|k}, x_{ss|k}$ are the estimated steady-states of the input and states which enable $y \to r_k$ asymptotically, r_k being the desired target at sample k.

Unbiased definitions of $u_{ss|k}, x_{ss|k}$ and their linear dependence on current disturbance estimate d_k and target r_k are well known in the literature e.g, [101], and can be defined for suitable K_{xr}, K_{ur} as follows.

$$\begin{bmatrix} x_{ss|k+i+1} \\ u_{ss|k+i} \end{bmatrix} = \begin{bmatrix} K_{xr} \\ K_{ur} \end{bmatrix} (r_{k+i+1} - d_k) \quad ....i \geq 0 \tag{4.3}$$

Remark 4.1 *This thesis focuses on infinite horizon algorithms due to their superior a priori stability properties. To simplify the presentation of the algebra, the disturbance estimate d_k is omitted from the equations hereafter; it is straightforward to include where required and so is included in some of the numerical illustrations.*

Thus, the predicted state and input evolution is conveniently captured by combining (4.2) and (4.3) to form a one-step ahead prediction model:

$$\begin{aligned} x_{k+1+i} &= \Phi x_{k+i} + [I - \Phi] K_{xr}(r_{k+1+i}) + B c_k \\ u_{k+i} &= -K x_{k+i} + [K K_{xr} + K_{ur}](r_{k+1+i}) + c_k \end{aligned} \tag{4.4}$$

4.2 OMPC dual-mode control law for constraint free case

This section shows how to derive a control law for an OMPC or dual-mode approach, including future information about target changes.

4.2.1 Autonomous model for predictions with a time-varying target

In order to derive a control law for a dual-mode for the unconstrained case with the inclusion of advance information about the targets, it is convenient to construct an equivalent state-space model which incorporates the predictions of (4.4) by adding the d.o.f. $\underset{\to k}{c}$ as additional states. It is also necessary to incorporate information about future target values $\underset{\to k+1}{r}$, so they become states in an augmented model.

The explicit assumption in this section is that the target is known n_a steps ahead and is assumed to be constant thereafter; hence ($r_{k+na+i} = r_{k+na}, i \geq 0$). A transition matrix linking the future target values can be constructed as follows.

$$\underbrace{\begin{bmatrix} r_{k+2} \\ r_{k+3} \\ \vdots \\ r_{k+n_a} \\ r_{k+n_a+1} \end{bmatrix}}_{\underset{\rightarrow k+2}{r}} = \underbrace{\begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & : \\ : & : & : & : & : \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & \cdots & I \end{bmatrix}}_{D_R} \underbrace{\begin{bmatrix} r_{k+1} \\ r_{k+2} \\ \vdots \\ r_{k+n_a-1} \\ r_{k+n_a} \end{bmatrix}}_{\underset{\rightarrow k+1}{r}} \tag{4.5}$$

where the notation $\underset{\rightarrow k+1}{r}$ means the future value of r_k.

Similarly, a transition matrix can be used to capture the future values of the d.o.f. $\underset{\rightarrow k}{c}$ for each sample within the prediction such that ($c_{k+n_c+i} = 0, i \geq 1$), given that:

$$\underbrace{\begin{bmatrix} c_{k+1} \\ c_{k+2} \\ \vdots \\ c_{k+n_c-1} \\ c_{k+n_c} \end{bmatrix}}_{\underset{\rightarrow k+1}{c}} = \underbrace{\begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & : \\ : & : & : & : & : \\ : & : & : & : & I \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{D_C} \underbrace{\begin{bmatrix} c_k \\ c_{k+1} \\ c_{k+2} \\ \vdots \\ c_{k+n_c-1} \end{bmatrix}}_{\underset{\rightarrow k}{c}} \tag{4.6}$$

It is convenient to combine the update equations (4.4-4.6) in a single autonomous model with an augmented state, giving the form:

$$Z_k = [x_k^T, \underset{\rightarrow k}{c}^T, \underset{\rightarrow k+1}{r}^T]^T; \qquad Z_{k+1} = [x_{k+1}^T, \underset{\rightarrow k+1}{c}^T, \underset{\rightarrow k+2}{r}^T]^T \tag{4.7}$$

The prediction model is given as:

$$Z_{k+1} = \Psi Z_k; \quad \text{where} \quad \Psi = \begin{bmatrix} \Phi & [B, 0, 0, ...] & [(I - \Phi)K_{xr}, 0, .., 0] \\ 0 & D_C & 0 \\ 0 & 0 & D_R \end{bmatrix} \tag{4.8}$$

4.2.2 Proposed Performance index with future target information

For a time-varying target, it is also convenient to incorporate changes in the target into the cost function of (2.31) as this is a precursor to the systematic design of a default FF compensator. An obvious change to this performance index is:

$$J = \sum_{i=1}^{\infty} \left\{ (x_{k+i+1} - x_{ss|k+i+1})^T Q (x_{k+i+1} - x_{ss|k+i+1}) + (u_{k+i} - u_{ss|k+i})^T R (u_{k+i} - u_{ss|k+i}) \right\} \tag{4.9}$$

where $u_{ss|k}, x_{ss|k}$ are the estimated steady-states of the input and states which enable $y \to r_k$ asymptotically, r_k being the desired target at sample k.

Summary: The closed-loop paradigm prediction structure of (4.2) is deployed as this is known to offer numerous numerical advantages [146] and indeed is now commonly adopted. Moreover, building on the performance index of (4.9) provides unique insights that are unavailable with the more conventional open-loop prediction formats.

Now, in order to define a suitable unbiased performance index which includes future information about set point changes, it is convenient to describe the deviation from the state in terms of the augmented state Z_k as illustrated below:

$$x_k - x_{ss|k} = \underbrace{\left[I, 0, .., 0\right] - \left[0 \quad 0 \quad [K_{xr}, 0, 0, ..0]\right]}_{K_{xss}} Z_k \tag{4.10}$$

Similarly, the deviation on the input can be described as:

$$u_k - u_{ss|k} = \underbrace{-\left[K, 0, .., 0\right] - \left[0 \quad 0 \quad [K_{ur}, 0, , ..., 0]\right]}_{K_{uss}} Z_k \tag{4.11}$$

Substituting (4.8), (4.10) and (4.11) into the performance index (4.9), one can express J in terms of the augmented state as:

$$J = \sum_{i=0}^{\infty} Z_{k+i}^T [\Psi^T K_{xss}^T Q K_{xss} \Psi + K_{zss}^T R K_{zss}] Z_{k+i} \tag{4.12}$$

Analogous to the illustration in Chapter 2, Section 2.8.3, the performance index (4.12) can be expressed in a simplified form,using the standard Lyapunov solution as:

$$J = Z_k^T S_z Z_k; \tag{4.13}$$

Remark 4.2 *The use of performance index (4.13) allows the user to formulate the explicit dependence of the control law on future target information.*

The matrix S_z can be decomposed into its individual block elements which show the links between the states x_k, $\underset{\rightarrow k+1}{r}$, $\underset{\rightarrow k}{c}$ within the cost function

$$S_z = \begin{bmatrix} S_x & S_{xc} & S_{xr} \\ S_{xc}^T & S_c & S_{cr} \\ S_{xr}^T & S_{cr}^T & S_r \end{bmatrix} \tag{4.14}$$

We use the decomposition of (4.14) to expand (4.13) as:

$$J = x_k^T S_x x_k + 2x_k^T S_{xc} \underset{\rightarrow k}{c} + \underset{\rightarrow k}{c}^T S_c \underset{\rightarrow k}{c} + \underset{\rightarrow k+1}{r}^T S_r \underset{\rightarrow k+1}{r} + 2x_k^T S_{xr} \underset{\rightarrow k+1}{r} + 2\underset{\rightarrow k}{c}^T S_{cr} \underset{\rightarrow k+1}{r} \tag{4.15}$$

However, within the optimisation of J, one can ignore the terms based on S_x, S_{xr} as these contain no d.o.f. and hence:

$$\arg\min_{\underset{\rightarrow k}{c}} J \equiv \arg\min_{\underset{\rightarrow k}{c}} \{ \underset{\rightarrow k}{c}^T S_c \underset{\rightarrow k}{c} + 2\underset{\rightarrow k}{c}^T S_{cr} \underset{\rightarrow k+1}{r} + 2x_k^T S_{xc} \underset{\rightarrow k}{c} \} \tag{4.16}$$

4.2.3 The final OMPC control law

Now, building on the corresponding performance J of (4.16), it is straightforward to define a control law by minimising the J w.r.t $\underset{\rightarrow k}{c}$, given that:

$$\frac{\mathrm{d}J}{\mathrm{d}\underset{\rightarrow k}{c}} = 0 \Rightarrow \underset{\rightarrow k}{c} = -S_c^{-1}[S_{cr} \underset{\rightarrow k+1}{r} + S_{xc} x_k]. \tag{4.17}$$

$$\underset{\rightarrow k}{c} = P_r \underset{\rightarrow k+1}{r} - S_c^{-1} S_{xc} x_k. \tag{4.18}$$

where $P_r \underset{\rightarrow k+1}{r} = -S_c^{-1} S_{cr} \underset{\rightarrow k+1}{r}$ is the feed-forward term. This control law depends on the future target information and the current state.

Remark 4.3 *By definition the optimal behaviour is given from $u_{k+i} - u_{ss|k+i} = -K(x_{k+i} - x_{ss|k+i})$, so, by definition, the optimal value of c_k must be zero, therefore $S_{xc} = 0$.*

Thus, the optimum $\underset{\rightarrow}{c}$ can be defined as:

$$\underset{\rightarrow k}{c} = P_r \underset{\rightarrow k+1}{r}. \tag{4.19}$$

Hence, the feed-forward term, in the unconstrained case, is given by P_r and the dependence on $\underset{\rightarrow k+1}{r}$ is explicit.

Corollary 4.1 *The summation of the block elements of the matrix P_r must be equal to zero. This follows immediately from the observation that if the future target is constant then the optimum unconstrained value is $c_k = 0$ and hence $P_r \underset{\rightarrow k+1}{r} = 0$. This reinforces the message that the feed-forward term is only active when advance information of* ***target changes*** *is used within the performance index.*

4.3 The effective use of advance knowledge for unconstrained systems

It was shown in the previous section that the results for optimising performance suggested that the optimum value of $\underset{\rightarrow k}{c}$ depends upon the future target values, through the feed-forward P_r, so the obvious inference is that one will obtain better performance by using this information and moreover using as large a n_a as possible. Surprisingly however, this intuitive expectation is incorrect. In fact, including the feed-forward term can cause a deterioration in closed-loop performance as will be shown below.

More specifically, this section seeks to provide more systematic guidance on how much feed-forward information is useful for improving output tracking performance and also, what constitutes too much feed-forward information which cannot be used effectively and thus can provide undesired results.

4.3.1 The impact of using advance target knowledge with optimal MPC

Most of the literature using OMPC algorithms ignores advance knowledge of target changes; that is, it tacitly assumes that, for the purpose of prediction and optimisation, $n_a = 1$ and

$r_{k+i} = r_{k+1}, \forall i > 0$. This also means that, within the predictions, $x_{ss|k}, u_{ss|k}$ are constant. Moreover, it can be shown [145] that in this case the optimum unconstrained choice of the d.o.f. is $\underset{\rightarrow k}{c} = 0$. This provides a useful observation which is a helpful insight for the control operator, as we develop the contributions of this thesis. When $n_a = 1$, that is no advance knowledge and the optimisation is reduced to minimising the weighted norm of the input perturbations $\underset{\rightarrow k}{c}$ so the magnitude of $\underset{\rightarrow}{c}$ is a direct indicator of the impact of the constraints on the input choices.

Summary: If $\underset{\rightarrow k}{c} = 0$, the constraints do not affect the choice of control inputs.

On the other hand, a key point to note from (4.17) is that, with the use of $n_a > 1$, the optimum values of input perturbations c_k are no longer zero, even in the unconstrained case, and now depend explicitly on the future target values as well as constraints. This is illustrated numerically in Section 4.4.

4.3.2 Determining the appropriate amount of advance knowledge

This subsection introduces two useful methods for selecting the optimum amount of advance knowledge for unconstrained systems. While these methods may seem somewhat simple or lacking in rigour, both offer the advantage of being easy to code/implement in practice which is a core aim of this thesis and moreover provides insights that allow the easy extension of the constraint handling case.

Method 1: Trial and error

This method uses trial and error to select the optimum amount of advance knowledge for unconstrained systems, introducing an effective procedure as follows:

1. For values n_a from 1 to n_y, simulate the process (for a specified target) and compute the cost J using the performance index (4.9) by summing the terms over the entire runtime (until all terms have converged to zero). Plot the cost J vs n_a .

2. Select the smallest n_a giving an acceptable J cost on the basis that a smaller n_a may be preferable if the loss in performance is minimal compared to a larger n_a.

Method 2: Alternative efficient proposal

The link with loop dynamics indicated in the controlled systems, suggests the potential for a more precise design guideline. It is desirable to obtain an output response in which the error before and after the set point change is balanced and the closed-loop settling time settles to within 10% of the steady-state. Therefore, one can argue from engineering common sense that the best choice of n_a, in the unconstrained case, lies between n_c and $n_c + n_s/2$ where n_s is the closed-loop settling time. Building on this concept, we introduce an alternative algorithm to determine the optimum amount of advance knowledge for unconstrained systems as follows:

Algorithm 4.1 *The algorithm of our proposal is listed as:*

1. *Determine the closed-loop settling-time n_s using a measure such as settling to within 10% of the steady-state.*

2. *Choose $n_a = n_a^* = n_c + n_s/2$.*

4.4 Numerical illustration of advance knowledge within the unconstrained case

This section discusses numerically by means of several examples, the embedding the advance knowledge, showing its effect on the performance and its effective use in the unconstrained OMPC approaches. Firstly, the section demonstrates the impact of the inclusion of advance information about set point changes ($n_a > 1$) on system performance. Moreover, the optimum answer is highly dependent on both the closed-loop dynamics and n_c. Secondly, this section evaluates the methods proposed in the previous section. Finally, the section ends with a comparison between the proposed methods followed by a section summary.

4.4.1 The impact of advance knowledge about the target changes on system performance

In order to illustrate the impact of advance knowledge on system performance, we will present two benchmark examples of different plant models. One example describes a four dimensional state, SISO system, the other example describes a large scale MIMO system as follows.

Example 1: SISO System

In this example, we consider the model of the water heater studied in [19]. The model matrices which describe the plant are given by:

$$A = \begin{bmatrix} 0.8351 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 & 0.5426 & 0 \end{bmatrix}, \quad D = 0 \tag{4.20}$$

In this system, the output is tracking a step input with an amplitude $r = 1$. The tuning parameter of the control input is $n_c = 2$, and the weighting matrices are $R = 0.1I$ and $Q = C^T C$. Both the inputs and states are unconstrained. By using the model (4.20), a discussion of three output tracking performances with and without the inclusion of advance knowledge and indeed the anticipation expected of a predictive control law deploying advance knowledge, are clearly demonstrated.

Figure 4.1 shows the tracking one response (lower plot) without advance knowledge ($n_a = 1$) and two responses with advance knowledge ($n_a = 5$ (upper right plots) and $n_a = 15$ (upper left plot)).

It is clear from Figure 4.1 that, for $n_a = 1$, the input perturbation term c_k is zero. However, for $n_a = 5$ and $n_a = 15$, c_k is non-zero during transients. It is also clear that the maximum input of the upper right plot remains far closer to the input steady-state than that of the lower one. Moreover, it is clear that at set point changes, the top plot anticipates more than the lower one and the upper right plot provides a faster output response and less aggressive initial control moves than the lower one.

Raising the advance knowledge to a very large value provides a slow drift (unnecessary anticipation) before moving quickly near the time of th significant set point change as shown in the upper left plot, so that the excessive use of advance knowledge ($n_a = 15$) in conjunction with a low $n_c = 2$ is not useful.

In this particular case, it might be argued whether the J is larger or not. This is proved, as shown in Table 4.1, where the plot with this advance knowledge provides a smaller J but the key point is that this much anticipation is undesirable, even if the technical performance improves slightly.

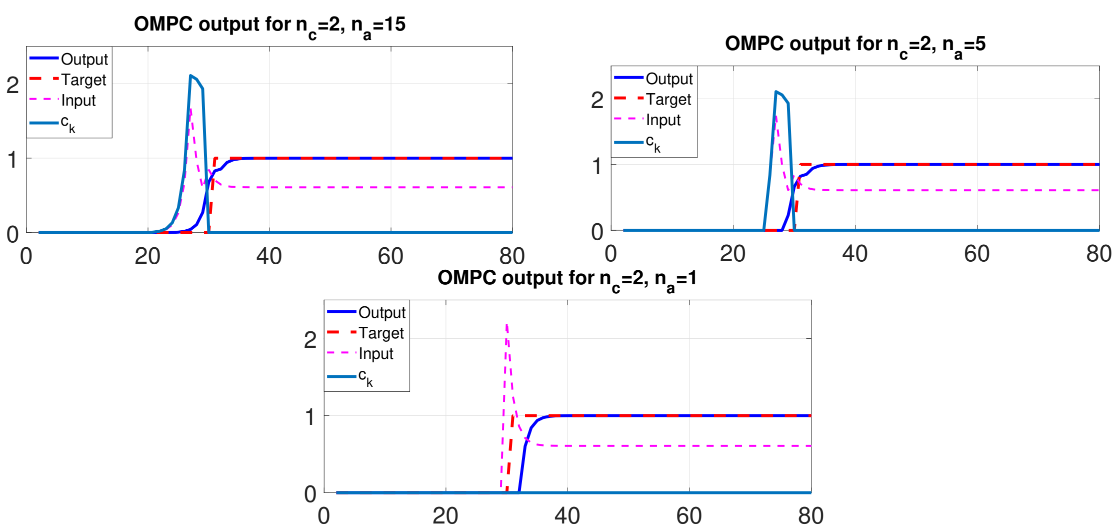

Figure 4.1: Closed-loop step responses for the SISO system (4.20) with $n_a = 1, 5$ and 15.

Example 2: the MIMO System

In this example, we consider the compressor model, studied in [19]. The discrete time transfer function of the model is given by:

$$G_{comp} = \begin{bmatrix} \frac{10^{-4}z^{-14}(0.7619z^{-1}+0.7307z^{-2})}{1-1.8806z^{-1}+0.8819z^{-2}} & \frac{0.022z^{-1}}{1-0.9761z^{-1}} \\ \frac{10^{-2}z^{-6}(0.7619z^{-1}+0.7307z^{-2})}{1-1.8534z^{-1}+0.8598z^{-2}} & \frac{10^{-2}z^{-19}(0.7619z^{-1}+0.7307z^{-2})}{1-1.2919z^{-1}+0.306z^{-2}} \end{bmatrix} \quad (4.21)$$

The model consists of 59 states with two manipulated variables and two controlled outputs, The matrices, A, B, C and D, which describe the model of the compressor are described in Appendix B.

Here both outputs which track two step inputs have an amplitude of $r_1 = 1.0,\ r_2 = 0.5$. The tuning parameters are $n_c = 3,\ Q = C^T C$ and the weighting factor of $R = diag(0.01, 0.01)$. Both inputs and states are unconstrained.

Again, we will discuss the tracking performance of three output responses with and without the inclusion of advance knowledge and indeed anticipated predictive control law's deployment of advance knowledge, using the corresponding model

Figure 4.2 shows the output tracking one response (left lower plot) without advance knowledge, ($n_a = 1$) and two responses with advance knowledge ($n_a = 15$ (upper right plots) and $n_a = 25$ (upper left plot)). Perturbations about the optimal are shown in the lower right plot.

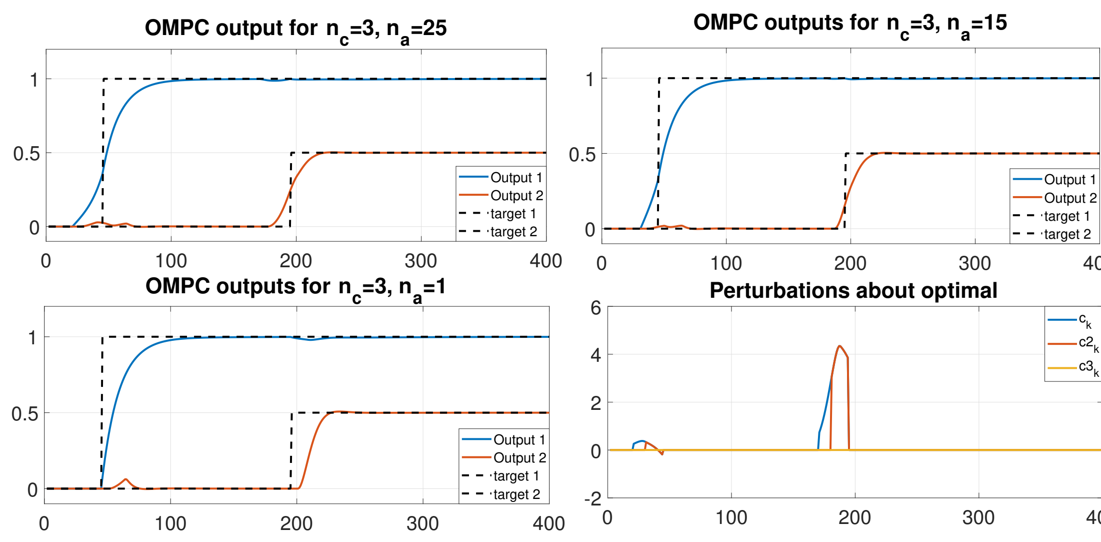

Figure 4.2: Closed-loop step responses for the compressor model with $n_a = 1, 15$ and 25.

Again it is clear from the lower right plot that with advance knowledge, c_k is non-zero during transients. It is also shown that the anticipation of the set point change in the upper plots and improvement in tracking are clear when the advance knowledge ($n_a = 15$ and $n_a = 25$)

is included, whereas there is no anticipation in the lower plot without advance knowledge. Regarding the interaction between the output variables, no down deviation is seen in the responses of the upper plots, although a slight deviation can be seen in the lower plot. It is clear, therefore, that using advance knowledge is beneficial. However, when the advance knowledge is too large the response shows unnecessary anticipation which is not useful in conjunction with a low $n_c = 3$ as we discussed previously for model (4.20).

Again, by calculating the cost J, it is seen that the numerical values for the cost J for the upper plot where advance knowledge is available is smaller than the values for the cost J for the lower plot where no advance knowledge exists. This is illustrated in tabular form as follows.

Table 4.1 shows the numerical values for the cost J for both systems (4.20) and (4.21), using the performance index of (4.9) for different advance knowledge values. This clearly demonstrates the superior performance of the proposed approach using advance knowledge.

	J with $n_a = 1$	J with $n_a = 5$	J with $n_a = 15$
System (4.20)	3.4966	0.7391	0.7115
	J with $n_a = 1$	J with $n_a = 15$	J with $n_a = 25$
System (4.21)	15.2337	9.2787	8.5190

Table 4.1: Performance indices for step changes in the target for system (4.20) and system (4.21).

It is shown in Table 4.1 that the cost function J with advance knowledge is slightly lower than that without advance knowledge. This provides further evidence of the advantages of using advance knowledge within the OMPC control law.

Summary: This subsection has shown how to implement advanced information within the new dual-mode or OMPC control law. It is clear that this information can be used systematically and critically affects behaviour by changing the perturbation term c_k.

4.4.2 Evaluation of the trial and error method and the proposed algorithm

This subsection evaluates both the trial and error method and proposed Algorithm 4.1 to select the best amount of advance knowledge, by means of the benchmark model which is introduced in [73]. Moreover, this model is used to demonstrate how easily the proposed methods can be applied to OMPC dual-mode approaches. The plant model matrices are given by:

$$A = \begin{bmatrix} 1.1 & 2.0 \\ 0 & 0.95 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 1 \end{bmatrix} \tag{4.22}$$

Example 1: Evaluation of the trial and error method

This example uses the model (4.22) to evaluate the trial and error method for selecting the appropriate advance knowledge values n_a.

A closed-loop simulation is performed with $n_c = 2$ and $Q = C^T C$ for various choices of the input weight matrix $R = 0.01I,\ 0.1I, 0.5I$ and I.

Figure 4.3 shows the corresponding cost function J versus the values of advance knowledge n_a for different input weights R.

It is clear that an appropriate choice of n_a depends on the tuning parameter of the feedback loop, R, as miscalculating this can have serious consequences regarding performance. Detuning (high R) tends to mean that a higher n_a is beneficial as the system is slow to respond and needs more anticipation to balance errors around the target changes. With more aggressive tuning, often a relatively small n_a is the optimum choice, as this prevents the system from anticipating the target changes earlier than necessary.

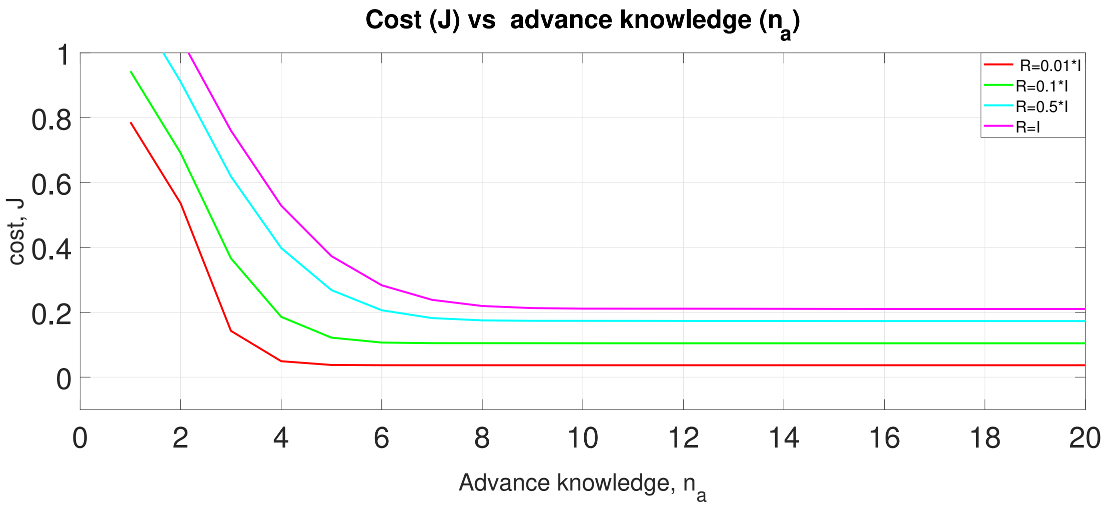

Figure 4.3: The cost J versus advance knowledge for the system (4.22) with $R = 0.01I, 0.1I, 0.5I$ and I.

Remark 4.4 *This subsection does not consider the impact of constraint handling so the reader is reminded that in practice, if $n_a \gg n_c$, then one is more likely to get infeasibility as there is poor matching between the d.o.f. and the change in the asymptotic steady-state (terminal constraints). Hence, irrespective of the outcomes of the trial and error method in the unconstrained case, in practice one may decide to choose a smaller n_a if the number of d.o.f. are limited to ease the constraint handling; this issue is discussed in more detail in the next section.*

Example 2: Evaluation of Algorithm 4.1

In this example, we will evaluate Algorithm 4.1 for selecting the appropriate advance knowledge , n_a values, to be used in the OMPC control design.

Firstly, we need to determine the settling time of the open loop response for the corresponding system. To achieve this, we then perform a simulation of the output dynamic response of system (4.22) with $n_c = 1$ and $Q = C^T C$ for various choices of the input weight matrix, with a unit step response, in order to determine the settling time for the corresponding open loop output response.

Figure 4.4 shows the open-loop response for the model for different control weights, $R = 0.01I,\ 0.1I, 0.5I$ and I.

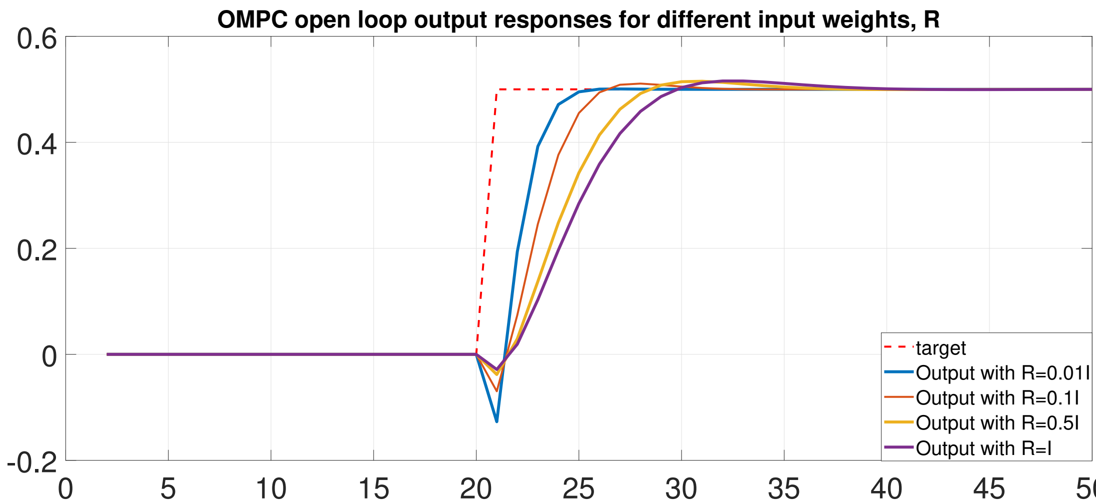

Figure 4.4: Open-loop step responses for system (4.22) with $R = 0.01I,\ 0.1I, 0.5I$ and I.

It can be seen from Figure 4.4 that the output responses are differ due to the effect of R on the system responses in which the system output response becomes slower as R increases; hence, various settling time values, n_s, of the system step response can be estimated as shown in Table 4.2.

R	$0.01I$	$0.1I$	$0.5I$	I
n_s	4	5	8	9

Table 4.2: The estimated settling time for the step response of system (4.22) for various R

Secondly, building on the results obtained from this simulation, we calculate the value of the advance knowledge, n_a, for a tuning parameter, $n_c = 2$ for different R, using the formula in Algorithm 4.1 as illustrated below in Table 4.3.

Finally, we select the computed values of n_a as an appropriate value for the system for associated R.

It is clear from Table 4.3 that the advance knowledge is high as the weight is high since increasing the weighting leads to slower responses and hence more future information about the target changes; in other words larger n_a is required. This suggests that the choice of appropriate advance knowledge is dependent on the system dynamics as well as the tuning parameters, n_c and R.

In order to demonstrate the benefits of using advance knowledge, we will perform a closed-loop simulation for the corresponding system (4.22). The tuning parameters are set at $n_c = 2$ and $Q = C^T C$ and $R = 0.01I$ with the target of $r_k = 0.5$. In this particular case, the value of n_a can be determined by using Algorithm 4.1, which is $n_a = 4$, as shown in Table 4.3.

Figure 4.5 shows the closed-loop response for the model without advance knowledge, $n_a = 1$ and with the selected advance knowledge value of $n_a = 4$.

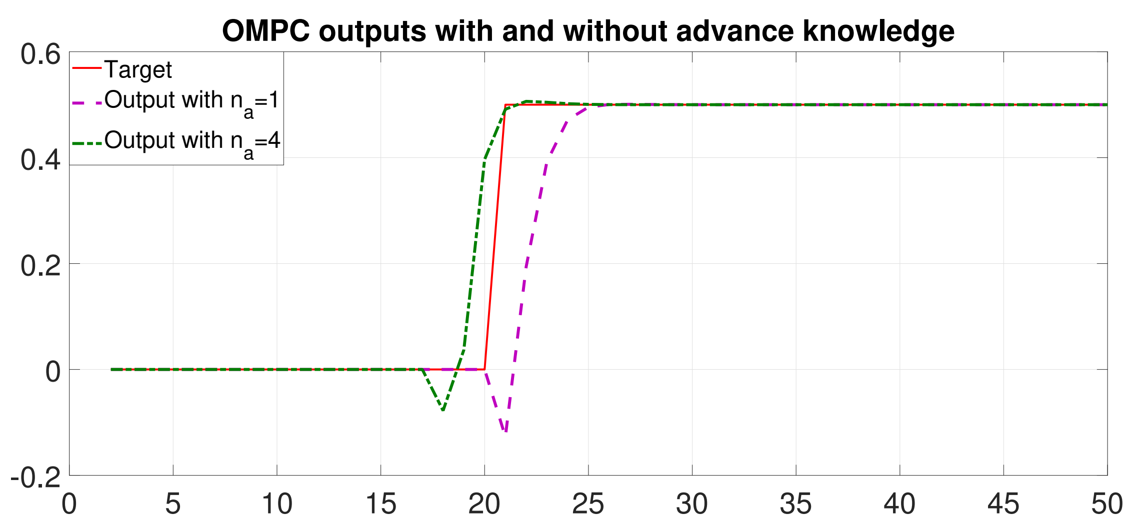

Figure 4.5: Closed-loop step responses of system (4.22) for $R = 0.01$ with $n_a = 1$ and $n_a = 4$.

Clearly, the output response with the obtained advance knowledge of $n_a = 4$ is faster than that without advance knowledge of $n_a = 1$, providing better tracking performance. Moreover, the response with advance knowledge provides a lower inverse action compared with that without advance knowledge.

R	n_s	$n_a^* = n_c + n_s/2$
0.01I	4	4
0.1I	5	5
0.5I	8	6
I	9	6

Table 4.3: The appropriate advance knowledge for system (4.22) with $n_c = 2$ for various R

Further evidence of the advantage of using advance knowledge can be shown by describing the performance indices for the corresponding model with and without advance knowledge in tabular form as shown in Table 4.4.

	J with $n_a = 1$	J with $n_a = 4$
System (4.20)	0.7862	0.0491

Table 4.4: Performance indices for step changes in the target for system (4.22)

It is clear that the performance index with advance knowledge is lower than that without advance knowledge. This indicates the usefulness of advance knowledge.

4.4.3 A comparison between the trial and error method and Algorithm 4.1

This subsection compares the two proposals discussed in the previous section by considering the model (4.22) with $n_c = 2$, $Q = C^T C$ and different weights, $R = 0.01I,\ 0.1I,\ 0.5I$ and I.

A comparison of the solutions obtained from trial and error method and Algorithm 4.1 is shown in Table 4.5 for four different choices of tuning.

System (4.22) $n_c = 2, R = 0.01I, n_s = 4$				Method 1	Alg.4.1
n_a	3	4	5	5	4
J	0.1427	0.0491	0.0375	0.0375	0.0491
System (4.22) $n_c = 2, R = 0.1I, n_s = 5$				Method 1	Alg.4.1
n_a	4	5	6	6	5
J	0.1859	0.1218	0.1067	0.1067 ‘	0.1218
System (4.22) $n_c = 2, R = 0.5I, n_s = 7$				Method 1	Alg.4.1
n_a	5	6	7	7	6
J	0.2684	0.2062	0.1821	0.1821	0.2062
System (4.22) $n_c = 2, R = I, n_s = 8$				Method 1	Alg.4.1
n_a	6	7	8	8	7
J	0.2832	0.2384	0.2194	0.2194	0.2384

Table 4.5: Variation in the performance indices for step changes in the target over the cost J for a range of n_a and comparison of the proposals obtained from Method 1 and Algorithm 4.1.

It is clear from this table that the application of Algorithm 4.1 is close enough to be useful and simple to select. In these two cases it is striking that the trial and error method suggests that a high n_a is better but, significantly, decreasing gains with relatively low n_a can be clearly observed, as shown in Table 4.5. The expected observation that increasing the weighting leads to slower responses and thus the benefits from a higher n_a are also apparent. Furthermore, it provides a sensible proposal for n_a with a minimum of computation.

4.4.4 Interim Summary

This section reveals the following insights:

1. It is clear that advance information can be used systematically and affects behaviour.

2. Up to a limit, choosing $n_a > 1$ improves performance compared to $n_a = 1$ but limited improvements for $n_a \gg n_c$.

3. It is possible to use the trial and error method to choose an optimum value of n_a for a given set point profile but this would be cumbersome to implement in practice whereas the simple guideline of $n_a^* \approx n_c + n_s/2$ suggested by Algorithm 4.1 is seen to be fairly effective in the unconstrained case and would be easier to deploy in general.

4.5 Constraint handling with advance knowledge within OMPC approaches

Before considering the constrained case, it is important to get the unconstrained case right as this will provide the foundation for including the constraints later. The previous section and some earlier work [166] gave an indication of a possible start point which is to determine the feed-forward term P_r separately from the on-line optimisation, to determine a choice of P_r which is known to be optimal in the unconstrained case; such a choice would depend on assumptions about the dynamics with the feedback loop and choices for both n_a and n_c.

4.5.1 The proposed performance index

The proposal hereafter is to embed the optimised feed-forward and then add d.o.f. around this for constraint handling, as required. The results are straightforward, but provided for completeness as they build a foundation for the next sections.

Theorem 4.1 *Minimisation of performance index (4.16) gives the same optimum $\underset{\rightarrow k}{c}$ as the following optimisation.*

$$\underset{\rightarrow}{\tilde{c}} = \arg\min_{\underset{\rightarrow}{\tilde{c}}} \quad \tilde{J} = \underset{\rightarrow k}{\tilde{c}} S_c \underset{\rightarrow k}{\tilde{c}}; \quad \underset{\rightarrow k}{c} = \underset{\rightarrow k}{\tilde{c}} + P_r \underset{\rightarrow k+1}{r} \tag{4.23}$$

Proof: A parametrisation of the input perturbations c_k, which includes the optimum feed-forward (4.17) and further d.o.f. for constraint handling can be defined as:

$$\underset{\rightarrow k}{c} = \underset{\rightarrow k}{\tilde{c}} + P_r \underset{\rightarrow k+1}{r} \tag{4.24}$$

Hence the term $\underset{\rightarrow k}{\tilde{c}}$ is a deviation from the unconstrained optimum in the tracking case.

The cost function is given by substituting (4.24) into (4.16). Hence

$$\tilde{J} \equiv [\underset{\to k}{\tilde{c}} + P_r \underset{\to k+1}{r}]^T S_c [\underset{\to k}{\tilde{c}} + P_r \underset{\to k+1}{r}] + 2[\underset{\to k}{\tilde{c}} + P_r \underset{\to k+1}{r}]^T S_{cr} \underset{\to k+1}{r} \tag{4.25}$$

$$\tilde{J} \equiv [\underset{\to k}{\tilde{c}}]^T S_c [\underset{\to k}{\tilde{c}}] + 2[\underset{\to k}{\tilde{c}}]^T S_c P_r \underset{\to k+1}{r} + 2[\underset{\to k}{\tilde{c}}]^T S_{cr} \underset{\to k+1}{r} \tag{4.26}$$

Substituting P_r of equation (4.17) into (4.26) yields

$$\tilde{J} \equiv [\underset{\to k}{\tilde{c}}]^T S_c [\underset{\to k}{\tilde{c}}] - 2[\underset{\to k}{\tilde{c}}]^T S_{cr} \underset{\to k+1}{r} + 2[\underset{\to k}{\tilde{c}}]^T S_{cr} \underset{\to k+1}{r} \tag{4.27}$$

$$\tilde{J} \equiv [\underset{\to k}{\tilde{c}}]^T S_c [\underset{\to k}{\tilde{c}}] \tag{4.28}$$

It is known that the unconstrained optimum choice is $\underset{\to k}{\tilde{c}} = 0$ and therefore the performance index must be a quadratic with no-affine term. Therefore, for some constant F:

$$\tilde{J} = [\underset{\to k}{\tilde{c}}]^T S_c [\underset{\to k}{\tilde{c}}] + F \tag{4.29}$$

which implies minimising J and minimising $\tilde{J}$ in the absence of constraints must give the same $\underset{\to k}{\tilde{c}}$ $\square$

4.5.2 The MCAS for time-varying targets

The MCAS for the dual-mode MPC has been defined in Subsection 2.8.5. It has been shown that this MCAS can be used for the scenarios of fixed targets. However, in the case of a time-varying target, we need to develop an alternative algorithm, which can handle the target variations. This algorithm will be illustrated next.

The predictions of (4.8) are defined as feasible if they satisfy the constraints for all future samples. Thus, for convenience, these constraints are represented as a set of matrix inequalities. Standard algorithms are available in the literature for determining these inequalities e.g.[52] or recent variants such as [114], which were discussed in Subsection 2.5.1. At this point, it is worth introducing the MCAS, such that the predictions of (4.8) satisfy the constraints.

Describing the constraints on the inputs (2.41) in terms of the autonomous model (4.8) gives

$$\begin{bmatrix} -K & [I,0,0,...] & [KK_{xr}+K_{ur},0,.,.,.,0] \\ K & -[I,0,0,...] & -[KK_{xr}+K_{ur},0,.,.,.,0] \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{u} \\ \underline{u} \end{bmatrix} \tag{4.30}$$

Similarly, the constraints on the states (2.44) can be described as:

$$\begin{bmatrix} C & 0 & 0 \end{bmatrix} Z_k \leq \overline{x} \tag{4.31}$$

Also, the limits on the target can be expressed as:

$$\begin{bmatrix} 0 & 0 & I \\ 0 & 0 & -I \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{r} \\ \underline{r} \end{bmatrix} \tag{4.32}$$

Combining all the sets (4.30-4.32) together, provides:

$$\underbrace{\begin{bmatrix} -K & [I,0,0,...] & [KK_{xr}+K_{ur},0,...,0] \\ K & -[I,0,0,...] & -[KK_{xr}+K_{ur},0,...,0] \\ C & 0 & 0 \\ 0 & 0 & I \\ 0 & 0 & -I \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} x_k \\ \underrightarrow{c}_k \\ \underrightarrow{r}_{k+1} \end{bmatrix}}_{Z_k} \leq \underbrace{\begin{bmatrix} \bar{u} \\ \underline{u} \\ \bar{x} \\ \bar{r} \\ \underline{r} \end{bmatrix}}_{f} \tag{4.33}$$

The state and input constraints can be satisfied if the target is sensible. Therefore, a critical requirement to ensure convergence is that the steady-state must lie in the interior of the constraints set for the inputs and states. This can be achieved using the following condition:

$$\begin{bmatrix} K_{ur} \\ K_{ur} \\ CK_{xr} \end{bmatrix} \overline{r} \leq \begin{bmatrix} \overline{u}-\epsilon \\ \underline{u}+\epsilon \\ \overline{x}-\epsilon \end{bmatrix}; \qquad \begin{bmatrix} K_{ur} \\ K_{ur} \\ CK_{xr} \end{bmatrix} \underline{r} \leq \begin{bmatrix} \overline{u}-\epsilon \\ \underline{u}+\epsilon \\ \overline{x}-\epsilon \end{bmatrix}; \quad \epsilon > 0 \tag{4.34}$$

In this manner, targets should be checked against these inequalities before deploying or setting limits on r. The inequalities of (4.33) can be described for each sample in a standard sample constraints form given that:

$$GZ_k \leq f \tag{4.35}$$

The key point here is to divide the sample constraints of (4.35) into two parts. One part ($G_1 Z_k \leq f_1$), which includes the constraint inequalities (4.30, 4.31), must be satisfied for every sample, k. The other part ($G_2 Z_k \leq f_2$), which includes the constraint inequalities (4.32), must be satisfied only for the first sample ($k = 0$). Therefore, we can re-arrange the constraints into the form:

$$\begin{bmatrix} G_1 \\ --- \\ G_2 \end{bmatrix} Z_k \leq \begin{bmatrix} f_1 \\ --- \\ f_2 \end{bmatrix}, \qquad \forall k \tag{4.36}$$

Because we assume that the value r is constant, the rows in G_2 will not be carried forward in any admissible iteration. Therefore, the admissible set which has been discussed previously in Subsection 2.8.5, can be modified to deal with this by starting the iteration on the rows in G_1, and thus excludes G_2 [145].

Thus, the MCAS can be defined in a standard form as:

$$F Z_k \leq t \tag{4.37}$$

where F and t are defined in equation (2.52).

It may be convenient to expand the MCAS (4.37) gives

$$\underbrace{\begin{bmatrix} M & N & V \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} x_k \\ \underset{\rightarrow k}{c} \\ \underset{\rightarrow k+1}{r} \end{bmatrix}}_{Z_k} \leq t \tag{4.38}$$

Now, the MCAS set can be described in a compact form as:

$$S_{MCAS} = \left\{ x : \exists \underset{\rightarrow k}{c} \quad s.t \quad M x_k + N \underset{\rightarrow k}{c} + V \underset{\rightarrow k+1}{r} \leq t \right\} \tag{4.39}$$

where M, N and V are suitable matrices and t is a vector of the limits.

Corollary 4.2 *An equivalent MCAS for control perturbations (4.24) with future target values is straightforward to construct. This follows directly from the substitution of (4.24) into (4.39) for suitable constants.*

$$M x_k + N \underset{\rightarrow k}{c} + V \underset{\rightarrow k+1}{r} \leq t \quad \Rightarrow \quad M x_k + N [\underset{\rightarrow k}{\tilde{c}} + P_r \underset{\rightarrow k+1}{r}] + V \underset{\rightarrow k+1}{r} \leq t \tag{4.40}$$

The MCAS equation of (4.40) can be described in a standard form as:

$$Mx_k + N \underset{\rightarrow k}{c} + V \underset{\rightarrow k+1}{r} \leq t \quad \Rightarrow \quad Mx_k + N \underset{\rightarrow k}{\tilde{c}} + \underbrace{[NP_r + V]}_{Q} \underset{\rightarrow k+1}{r} \leq t \tag{4.41}$$

or in a compact form:

$$Mx_k + N \underset{\rightarrow k}{c} + V \underset{\rightarrow k+1}{r} \leq t \quad \Rightarrow \quad Mx_k + N \underset{\rightarrow k}{\tilde{c}} + Q \underset{\rightarrow k+1}{r} \leq t \tag{4.42}$$

At this point, the MCAS set can be described in a standard form as:

$$S_{MCAS} = \{x : \exists \tilde{c}_k \;\; s.t \; Mx_k + N \underset{\rightarrow k}{\tilde{c}} + Q \underset{\rightarrow k+1}{r} \leq t\} \tag{4.43}$$

where $M,\ N$ and Q are suitable matrices and t is a vector of the limits.

Summary: Now, the constrained optimisation can focus on the computation of just $\underset{\rightarrow k}{\tilde{c}}$ as the bias term to deal with advance information systematically is fully embedded. Specifically, $\tilde{c}_k \neq 0$ will be required if the constraints are active and the magnitude of $\tilde{c}_k$ is an indicator of how far one is from the unconstrained optimal associated to the advance knowledge scenario.

4.5.3 Key Observations

The previous subsection has utilised the procedures required for constructing the corresponding MCAS with the following observations:

1. The MCAS shape and position changes as the target r_k changes (e.g. [137]).

2. The MCAS is based on an augmented state which includes the target steady-state and thus is flexible to changes in target and disturbance.

3. This target must be reachable and also feasible during transient with the given n_c.

4. To ensure convergence the limits on the target should be selected with care.

4.5.4 Constrained OMPC Dual mode algorithm

Now, we can define the proposed algorithm which handles both the constraints and advance information about the targets.

Algorithm 4.2 *The constrained OMPC algorithm with the systematic incorporation of advance knowledge can now be summarised as:*

$$\min_{\underset{\rightarrow}{\tilde{c}}} \quad \tilde{J} = \underset{\rightarrow k}{\tilde{c}}^T S_c \underset{\rightarrow k}{\tilde{c}} \quad s.t. \quad Mx_k + N \underset{\rightarrow k}{\tilde{c}} + Q \underset{\rightarrow k+1}{r} \leq t \tag{4.44}$$

The optimised $\tilde{c}_k$ is used in conjunction with (4.4 and 4.24) to determine u_k.

4.6 Numerical examples for the proposed constrained algorithm for a reachable target

In order to demonstrate the efficacy of Algorithm 4.2 that uses advance knowledge of target changes with the constraints in the optimisation problem (assuming feasible target), we will perform a closed-loop simulation for different plant models with advance knowledge using the following examples:

4.6.1 Example 1: Six state dimensional SISO system

In this example, we consider the coke furnace process whose linear discrete transfer function was introduced in [174]. The corresponding discrete state space model can be obtained giving the following system matrices:

$$A = \begin{bmatrix} 0.9048 & 0 & 0 & 0 & 0 & 0 \\ 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0000 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.25 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{4.45}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.4188 \end{bmatrix}, \tag{4.46}$$

The input is the fuel flow, u and the output is the outlet temperature, y.

A closed-loop simulation is performed for system (4.45) with tuning parameters: the control perturbations, $n_c = 2$, $R = 0.01I$, $Q = C^T C$, subject to the following input and state constraints:

$$-0.5 \leq u_k \leq 1.35 \quad -4 \leq y_k \leq 4 \tag{4.47}$$

The allowable target limits for this case are $-2 \leq r \leq 2$; hence a feasible target of $r = 1.0$ is introduced.

For this model dynamics, the appropriate advance knowledge value is $n_a = 7$ since the settling time, $n_s = 10$ and the control perturbation, $n_c = 2$. (see Algorithm 4.1).

Figure 4.6 shows the performance (upper left plot) for system (4.45) with advance knowledge of $n_a = 7$, and the control input evolution (upper right plot). The figure also shows the control perturbation (lower plot) for the OMPC algorithm.

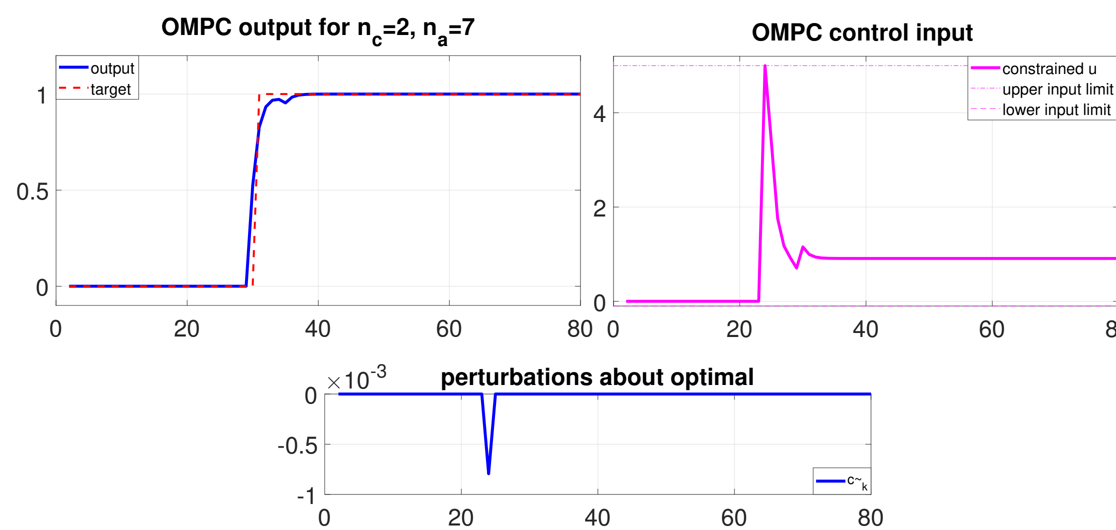

Figure 4.6: Closed-loop step responses for system (4.45) with $n_c = 2, \quad n_a = 7$.

It is shown in Figure 4.6 that Algorithm 4.2 provides effective control for the constrained system with advance knowledge of $n_a = 7$. Readers will note that the perturbations term $\underset{\rightarrow k}{\tilde{c}}$ (lower figure) is non-zero during the transients only, as expected. This is because the input constraints are active in the transients, as shown in the upper right plot. Moreover, the balance of the error around the set-point change (anticipation) is clear, which indicates the improvement in tracking system performance.

4.6.2 Example 2: Two state dimensional MIMO system

In this example, we consider the benchmark problem (double integrator), originally proposed in [86]. The matrices which describe the model of the plant are given by:

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}; \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.48}$$

A closed-loop simulation is performed for system (4.48) with the following tuning parameters: $n_c = 2$, $R = diag(0.1, 1)$, $Q = C^T C$, subject to the following input and state constraints:

$$\begin{bmatrix} -1 \\ -1 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} \tag{4.49}$$

For these constraints, the feasible target must lie within the following limits:

$$\begin{bmatrix} -1 \\ -0.25 \end{bmatrix} \leq r \leq \begin{bmatrix} 1 \\ 0.25 \end{bmatrix} \tag{4.50}$$

Here both states correspond to outputs which track the two step inputs with the targets $r_1 = 1.0,\ r_2 = 0.5$.

For this model dynamics, the appropriate advance knowledge value can be chosen to be equal to n_c. This means $n_a = 2$, since the settling time of the open-loop response, $n_s = 2$, is small (see Algorithm 4.1).

Figures 4.7 and 4.8 show the performance of the two outputs and control input evolutions for system (4.48), using advance knowledge. The first output tracks a reference amplitude of $r_1 = 1$ while the second output tracks a reference amplitude of $r_1 = 0.5$.

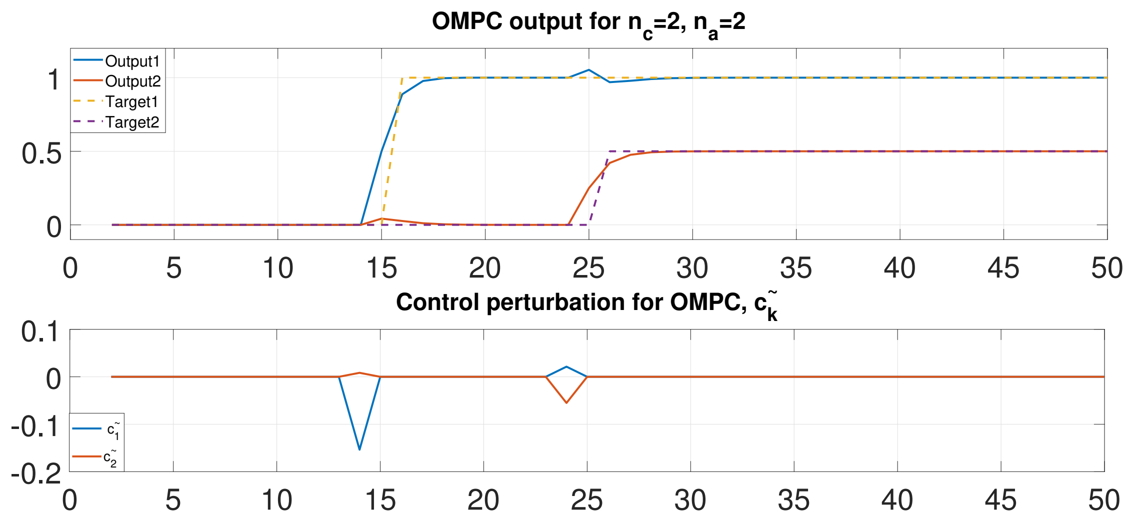

Figure 4.7: Closed-loop step responses for system (4.48) with $n_c = 2, \quad n_a = 2$.

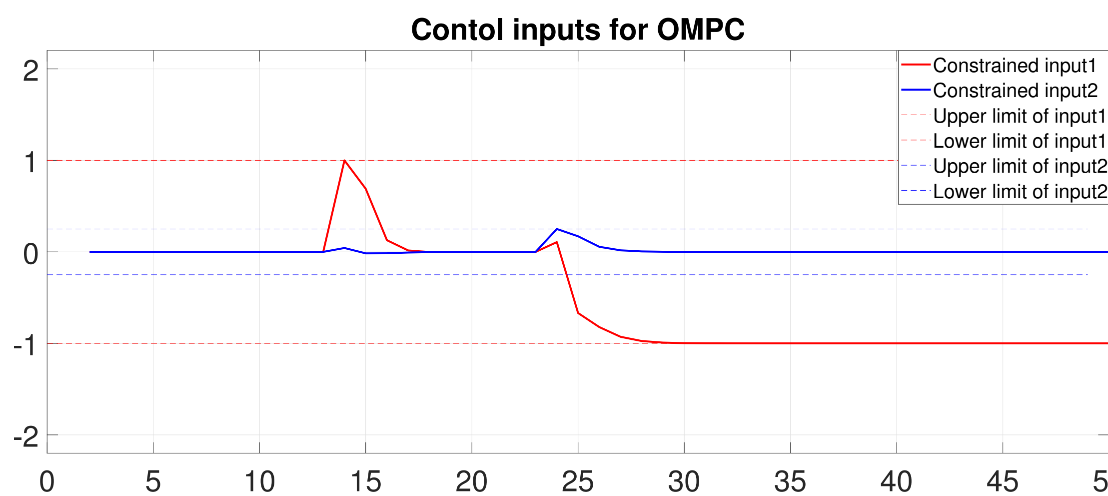

Figure 4.8: The evolution of the control inputs for system (4.48) with $n_c = 2, \quad n_a = 2$.

It is clear that Algorithm 4.2 provides effective control for the constrained MIMO system with advance knowledge ($n_a = 2$), giving tracking performance with reasonable anticipation and the outputs reach their steady-state for a short time, giving good tracking performance. Readers will note from Figure 4.7 that the perturbation variables $\underset{\rightarrow k}{\tilde{c}}$ (the lower panel) for both responses are non-zero during transients only and become zero for the long term, as expected. It will be noted that the perturbation term, $\underset{\rightarrow k}{c}$ is non-zero as expected from equation (4.24)

It can be also seen from Figure 4.8 that the constraints on both inputs, u_1 and u_2,are active in the transients; moreover, the steady-state value of the first control input, u_1, lies on the boundary of the input limits. This suggests that the control input can reach its steady-state, as it lies within the constraint limits.

Summary: The key point here is that, the default OMPC algorithm has the nice property that a choice of $\underset{\rightarrow}{c} = 0$ implies that the unconstrained optimal is feasible and thus one has a clear view of the impact of the constraints as there is a direct link to the magnitude of $\underset{\rightarrow}{c}$. However, including advance knowledge destroyed this link (see eqn.4.24). By re-parametrising the degrees of freedom in terms of $\underset{\rightarrow}{\tilde{c}}$, this nice property is recovered and, moreover, the nominal optimal solution, incorporating advance knowledge, becomes embedded within the predictions. Hence the required on-line optimisation, standard quadratic programming (QP), is solely dealing with constraint handling and not trying to achieve mixed objectives of performance optimisation and handling advance information alongside constraint handling.

4.7 Discussion on the use of advance knowledge with unconstrained and constrained cases

In this section, we discuss the difference between the unconstrained case and constrained case of the OMPC dual-mode, to assess the advance knowledge of future target information, as shown below.

For the unconstrained case, it is possible to choose the appropriate value of advance knowledge n_a, by using the trial and error method, but only to an extent. This is because $n_a >> n_c$ may yield little benefit. Moreover, in practice, it is convenient to select n_a, using the proposed Algorithm 4.1.

On the other hand, the inclusion of too large a n_a may not be helpful for the scenario in which constraints are active. This is due to a mismatch between the location of the degree of freedom and target change. Moreover, it can be argued that to ensure optimal behaviour, it is necessary to embed the default unconstrained feed-forward in the optimisation with a small value of n_a to ease the constraint handling.

4.8 Conclusion

This chapter has investigated the impact of advance knowledge about future target information on feed-forward design within predictive control. It is known that in principle advance information about target changes (and indeed measurable disturbances) can be included within predictive control optimisation, but it is less well known that the default incorporation of this information can lead to a degradation rather than an improvement in performance.

This chapter has shown, by illustration, that, in practice, it is often better to use only a subset of the future information available during MPC optimisation for each sample. Too little advance information will result in a delayed response, whereas too much advance information can lead to earlier than desired anticipation. However, a critical observation is that it is difficult to provide a theoretical result, and indeed we suspect that a generic, useful theoretical result will be impossible, with regard to the *optimal* amount of advance information to use as this varies with the systems, horizons, weights and constraints in a non-simple fashion.

Section 4.3 has proposed a simple algorithmic approach which allows the user to obtain a close to optimal answer regarding the amount of advance information, n_a, quickly and easily and thus in a manner useful to a field engineer who simply seeks an approximation. The examples show that the proposed approach gives sensible answers for infinite horizon or OMPC dual-mode MPC algorithms and has the advantage of being simple and pragmatic.

An argument is made that during constraint handling, it is better to construct predictions which embed the default *optimal unconstrained feed-forward* rather than enter the future target values directly. This ensures that the optimal behaviour is embedded and adds transparency to the role of the degrees of freedom. The efficacy and simplicity of this approach is demonstrated. Moreover, this work has focussed on simple step changes in the target as these typically capture the core insights. However, one challenge is how to deal with the scenario of excessively large changes in the target; that is, the target may be unreachable during the transient or steady-state. This will be discussed in the next chapter as another of the thesis contributions.

Chapter 5

FEASIBILITY WITH ADVANCE KNOWLEDGE WITHIN OMPC TRACKING

In practice, there is a scenario in which the constraints are active in the steady-state due to rapid set point changes. In this scenario, the true set point may become unreachable. Recent work has proposed the use of an artificial target which is reachable, but the challenge here is how to compute and select this target and moreover, how to incorporate it into the optimisation on the MPC. Therefore, it is logical to devise a novel algorithm that computes and chooses an artificial target and incorporates it into the performance index using dual-mode OMPC algorithms to show the impact of the terminal constraints on feasibility. This will be discussed in this chapter in relation to the constrained case.

Section 5.1 discusses how to ensure feasibility and stability by incorporating an artificial target into the OMPC optimisation problem. Section 5.2 proposes an input parametrisation which is appropriate for dealing with unreachable targets. Section 5.3, studies several performance indices and proposes a modified performance index that handles unreachable targets. Section 5.4, derives an autonomous dynamic model for unreachable targets. Section 5.5 discusses constraint handling with the unreachable target scenario and proposes an efficient algorithm which is suitable for this scenario. How to guarantee feasibility and performance is introduced in Section 5.6. Several observations are listed in Section 5.7, while Section 5.8 demonstrates the proposed algorithms via *Matlab* simulations. Finally, the conclusions are presented in Section 5.9

5.1 Unreachable targets and advance knowledge

The previous chapter focused on the effective use of advance knowledge when target changes are feasible so that optimisation (4.44) always offers a solution. This chapter extends the discussion to scenarios where infeasibility occurs; that is, where the change in the steady-state $x_{ss|k}, u_{ss|k}$ is too rapid, so the prediction class (4.1) is insufficiently large to meet the constraints. Infeasibility can take two common forms:

1. Transient infeasibility; that is, the target is reachable asymptotically but a far larger n_c is required [125] to find a feasible solution. Assuming that n_c cannot be increased, an alternative algorithm is needed to maintain feasibility and convergence. A common proposal [143, 137, 86, 156]) is to include extra degrees of freedom (d.o.f) that capture changes in the steady-state. A contractive constraint may be deployed to ensure convergence [23]. This is not pursued in this thesis.

2. Persistent infeasibility or so-called unreachable targets [156, 126]. In this case, the target cannot be reached, even asymptotically, without violating some constraints and thus an alternative parametrisation allowing changes to the target steady-state is needed again, along with a modified objective.

A key point to note here is that the majority of the work in the literature tackling these two issues assumes that $n_a = 1$; in this chapter, proposals are made which embed advance knowledge (i.e. $n_a > 1$) while also taking account of transient or permanent infeasibility and, moreover, while retaining a simple QP based optimisation.

Remark 5.1 *Reference governor approaches [50, 3] have analogies to both of the above forms, as a governor deploys a transient change in the target to maintain feasibility. However these will not be discussed further as their focus is on simplicity, but at a cost to performance.*

5.2 Input parametrisation for unreachable targets

In the case where the asymptotic target is unreachable, then the input parametrisation of (4.1) is invalid; that is, infeasible. The proposal here is to replace this parametrisation with:

$$\begin{cases} x_{k+1} - x_{ss|k} = A(x_k - x_{ss|k}) + B(u_k - u_{ss|k}); \quad u_k - u_{ss|k} = -K(x_k - x_{ss|k}) + c_k, \;\; k < n_c \\ x_{k+1} - x_{ss|k} = A(x_k - x_{ss|k}) + B(u_k - u_{ss|k}); \quad u_k - u_{ss|k} = -K(x_k - x_{ss|k}) + c_\infty, \; k \geq n_c \end{cases} \tag{5.1}$$

Lemma 5.1 *The inclusion of the term c_∞ within the input parametrisation of (5.1) leads to a constant offset between the predicted steady-state output and the desired target.*

Proof: Substitute the asymptotic input parametrisation of (5.1) into the model dynamics (2.1, 2.32). It is clear that if $c_\infty = 0$, there is no offset and hence:

$$\lim_{k\to\infty} x_k = x_{ss} = K_{xr} r_{k+n_a} \quad \Rightarrow \quad \lim_{k\to\infty} y_k = r_{k+n_a} \tag{5.2}$$

Using superposition one can then determine that, with (5.1):

$$\lim_{k\to\infty} y_k = r_{k+n_a} + \delta y_\infty; \quad \delta y_\infty = \underbrace{[C(I-\Phi)^{-1}B]^{-1}}_{G_\infty} c_\infty \qquad \square \tag{5.3}$$

Corollary 5.1 *The inclusion of c_∞ is equivalent to deploying an artificial target $\hat{r}$ which is deviated from the true target by δy_∞. Hence, one can also find an equivalent c_∞ for a specified artificial target $\hat{r}$ as follows.*

$$c_\infty = G_\infty^{-1}(\hat{r}_k - r_{k+n_a}) \tag{5.4}$$

Remark 5.2 *Although notionally denoted as dual-mode due to the input selection with $c_{k+i} = c_\infty, i \geq n_c$, one may view predictions (5.1) as having a parallel dual-mode due to the target dynamics whereby $r_{k+i} = r_{k+n_a}, i \geq n_a$.*

5.3 Performance indices for unreachable targets

This section introduces several performance indices which can be implemented in scenarios related to unreachable targets as follows.

5.3.1 Performance index with slack variables

It was shown in the previous chapter that the cost function with future target information can be described as:

$$\arg\min_{\underset{\rightarrow k}{c}} J \equiv \arg\min_{\underset{\rightarrow k}{c}} \{ \underset{\rightarrow k}{c}^T S_c \underset{\rightarrow k}{c} + 2 \underset{\rightarrow k}{c}^T S_{cr} \underset{\rightarrow k+1}{r} \} \tag{5.5}$$

It was also shown that the minimisation of performance index (5.5) gives the same optimum $\underset{\rightarrow k}{c}$ as the following optimisation:

$$\underset{\rightarrow}{\tilde{c}} = \arg\min_{\underset{\rightarrow}{\tilde{c}}} \quad \tilde{J} = \underset{\rightarrow k}{\tilde{c}} S_c \underset{\rightarrow k}{\tilde{c}}; \quad \underset{\rightarrow k}{c} = \underset{\rightarrow k}{\tilde{c}} + P_r \underset{\rightarrow k+1}{r} \tag{5.6}$$

It is convenient to choose an arbitrary cost function, such as including a slack variable c_∞, which can be derived as follows.

Lemma 5.2 *Minimising the performance index (5.5) with input parametrisation (5.1) and $n_a = 1$ gives the same optimising values for c_k as minimising the following cost:*

$$J_c = \underset{\rightarrow k}{c}^T S_c \underset{\rightarrow k}{c} + \sum_{i=1}^{\infty} c_\infty^T S c_\infty \tag{5.7}$$

Proof: Minimising the true performance index J was shown earlier (Remark 4.2) to be equivalent to minimising:

$$J_c = \sum_{i=0}^{\infty} c_{k+i}^T S_c c_{k+i} \tag{5.8}$$

Then, noting that in effect the parametrisation (5.1) implies $c_{k+n_c+i} = c_\infty, i > 0$ the result drops out. The $\underset{\rightarrow k+1}{r}$ has been excluded from J_c because here $n_a = 1$. $\square$

Corollary 5.2 *By combining the input parametrisation of (5.1) with the observation of equation (5.6) and Lemma 5.2 one can form an equivalent cost function for $n_a > 1$ of the form:*

$$\tilde{J}_c \equiv \underset{\to k}{\tilde{c}}^T S_c \underset{\to k}{\tilde{c}} + \sum_{i=1}^{\infty} c_\infty^T S c_\infty \tag{5.9}$$

Equivalent means that the optimum control law obtained from minimising $\tilde{J}_c$ is the same as the optimal control law obtained from minimising J_c.

5.3.2 Performance index with steady-state offset

It is well recognised [143, 126] that the performance index of (5.9) is not useful in itself because whenever $c_\infty \neq 0$ this J_c is unbounded and hence minimising J_c is equivalent to minimising the offset component of $c_\infty^T S c_\infty$. Indeed, one could choose simply to do this, but such an objective would effectively ignore the impact of transient behaviour on overall performance and thus may lead to relatively poor decisions. Consequently, a performance index is necessary, which captures the following requirements:

1. Has an objective measure of transient performance.

2. Is always feasible and thus includes the d.o.f. c_∞ to allow deviation from unreachable asymptotic targets.

3. Includes advance information about target changes.

The key proposal here is to build on the performance index of (5.9) which already includes transient performance and implicitly includes information about advance knowledge through the deployment of $\tilde{c}_k$. However, we desire a reduced emphasis on the asymptotic predicted error so that this does not swamp the transient terms.

5.3.3 The proposed performance index

Building on the discussion outlined in the previous section, it is convenient to propose a performance index J_p which provides a balance between transient behaviour and expected asymptotic offset. This can be defined as:

$$J_p = W_1(c_\infty^T S c_\infty) + \underset{\rightarrow k}{\tilde{c}}^T S_c \underset{\rightarrow k}{\tilde{c}} \tag{5.10}$$

where W_1 is a scalar weighting between the transient and asymptotic to be selected. Here, the term $c_\infty^T S c_\infty$ penalises asymptotic offset and the term $\underset{\rightarrow k}{\tilde{c}}^T S \underset{\rightarrow k}{\tilde{c}}$ penalises transient performance, including information on $\underset{\rightarrow k+1}{r}$. The scalar weighting W_1 allows the user to determine the emphasis they wish to place on each term.

Summary: In summary, a key aim of this section was to show how an additional d.o.f. can be added which allows the MPC algorithm to cater for unreachable points. Moreover, this d.o.f. was added in such a way that gives clarity to the impact of constraints.
Nevertheless, a more significant contribution is to show how optimal trajectories which include advance knowledge of the targets can also be embedded effectively. The $\tilde{c}$ terms indicate the deviation from the unconstrained optimal, with advance knowledge, during transients and the c_∞ term gives the steady-state deviation from the unreachable target.

5.4 Autonomous model for predictions with unreachable targets

Similarly, as presented in the previous chapters, one can define the predictions such that the augmented state includes an additional term as compared to (4.8); that is the term $c_{k+n_c+i} = c_\infty, i \geq 0$. Hence, the autonomous model can be derived as follows.

Consider a closed-loop prediction model:

$$\begin{aligned} x_{k+1} &= \Phi x_k + [I - \Phi]K_{xr}(r_{k+1}) + Bc_k \\ u_k &= -Kx_k + [KK_{xr} + K_{ur}](r_{k+1}) + c_k \end{aligned} \tag{5.11}$$

A transition matrix can be used to capture the d.o.f. $\underset{\rightarrow k}{c}$ for each sample within the prediction and hence:

$$\underbrace{\begin{bmatrix} c_{k+1} \\ \vdots \\ \vdots \\ c_{k+n_c} \\ c_{k+n_c+1} \end{bmatrix}}_{\underset{\rightarrow k+1}{c}} = \underbrace{\begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & : \\ : & : & : & : & : \\ 0 & 0 & 0 & \cdots & I \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{D_C} \underbrace{\begin{bmatrix} c_k \\ c_{k+1} \\ \vdots \\ \vdots \\ c_{k+n_c} \end{bmatrix}}_{\underset{\rightarrow k}{c}} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ I \end{bmatrix}}_{E_C} c_\infty \tag{5.12}$$

For convenience, we can define an additional perturbation in a compact form as:

$$\underset{\rightarrow k+1}{c} = D_C \underset{\rightarrow k}{c} + E_C c_\infty \tag{5.13}$$

Substituting equation (5.13) into equation (5.11), one can rewrite the closed-loop dynamics in the autonomous model form as:

$$Z_{k+1} = \Psi Z_k \tag{5.14}$$

where

$$\Psi = \begin{bmatrix} \Phi & [B, 0, \cdots, 0] & 0 & [(I-\Phi)K_{xr}, 0, \cdots, 0] \\ 0 & D_c & E_c & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & D_R \end{bmatrix}; \quad Z_k = [x_k^T, c_k^T, c_\infty, \underset{\rightarrow k+1}{r}^T]^T$$

where D_c, E_C and D_R are suitable shift matrices.

The input can also be described in terms of the autonomous model as:

$$u_k = \underbrace{\begin{bmatrix} -K & [I, 0, ..., 0] & 0 & [KK_{xr} + K_{ur}, 0, \cdots, 0] \end{bmatrix}}_{K_Z} Z_k \tag{5.15}$$

or in a compact form as:

$$u_k = K_Z Z_k \tag{5.16}$$

5.5 Constraint handling for unreachable targets

It is worth introducing an MCAS for the unreachable target such that the dynamics of (5.15) satisfy the constraints. This can be defined as follows.

It is convenient to express both the input and the state constraints in an inequality form as:

$$\underline{u} \leq u_k \leq \bar{u}; \qquad K_{xmax}x_k \leq \bar{x} \tag{5.17}$$

The constraints on the steady-state values of the control inputs and the states can be described as:

$$\underline{u} \leq K_{ur}(r_{k+na} + c_\infty) \leq \bar{u}; \quad K_{xr}(r_{k+na} + c_\infty) \leq \ \bar{x} \tag{5.18}$$

The constraints of (5.17) and (5.18) can be described in terms of the autonomous model as:

$$\begin{bmatrix} -K & [I,0,...,0] & 0 & [KK_{xr}+K_{ur},0,..,0] \\ K & [I,0,...,0] & 0 & -[KK_{xr}+K_{ur},0,..,0] \\ K_{xmax} & 0 & 0 & 0 \\ 0 & 0 & K_{ur} & [0,..0,K_{ur}] \\ 0 & 0 & -K_{ur} & [0,..,0,-K_{ur}] \\ 0 & 0 & K_{xr} & [0,..,0,K_{xr}] \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{u} \\ \underline{u} \\ \bar{x} \\ \bar{u} \\ \underline{u} \\ \bar{x} \end{bmatrix} \tag{5.19}$$

The asymptotic target limits can be given as:

$$\begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & -I \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{r} \\ \underline{r} \end{bmatrix} \tag{5.20}$$

Combining (5.19) and (5.20) together, provides

$$\underbrace{\begin{bmatrix} -K & [I,0,0,...] & 0 & [KK_{xr}+K_{ur},0,\cdots,0] \\ K & -[I,0,0,...] & 0 & -[KK_{xr}+K_{ur},0,\cdots,0] \\ 0 & 0 & K_{ur} & [0,\cdots,0,K_{ur}] \\ 0 & 0 & -K_{ur} & [0,\cdots,,0,-K_{ur}] \\ K_{xmax} & 0 & 0 & 0 \\ 0 & 0 & K_{xr} & [0,\cdots,0,K_{xr}] \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & -I \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} x_k \\ \underset{\rightarrow k}{c} \\ c_\infty \\ \underset{\rightarrow k+1}{r} \end{bmatrix}}_{Z_k} \leq \underbrace{\begin{bmatrix} \bar{u} \\ \underline{u} \\ \bar{u} \\ \underline{u} \\ \bar{x} \\ \bar{x} \\ \bar{r} \\ \underline{r} \end{bmatrix}}_{f} \tag{5.21}$$

It is straightforward to apply the admissible set algorithm discussed in Section 4.5.2 to find an invariant/admissible set of the following format:

$$FZ_k \leq t \tag{5.22}$$

where F and t are defined in equation (2.52).

The inequality of (5.22) can be expanded to:

$$\underbrace{\begin{bmatrix} M & N & T & V \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} x_k \\ \underset{\rightarrow k}{c} \\ c_\infty \\ \underset{\rightarrow k+1}{r} \end{bmatrix}}_{Z_k} \leq t \tag{5.23}$$

Now, the MCAS set can be described in an expanded form as:

$$S_{MCASU} = \{x : \exists (c_k, c_\infty) \;\; s.t \; Mx_k + N\underset{\rightarrow k}{c} + Tc_\infty + V\underset{\rightarrow k+1}{r} \leq t\} \tag{5.24}$$

where M, N, T and V are suitable matrices and t is a vector of limits.

Corollary 5.3 *An equivalent MCAS for control perturbations (5.6) with future target values is straightforward to construct. This follows directly from the substitution of (5.6) into (5.24) for suitable constants:*

$$Mx_k + N\underset{\rightarrow k}{c} + Tc_\infty + V\underset{\rightarrow k+1}{r} \leq t \quad \Rightarrow \quad Mx_k + N\underset{\rightarrow k}{\tilde{c}} + Tc_\infty + \underbrace{[NP_r + V]}_{Q}\underset{\rightarrow k+1}{r} \leq t \tag{5.25}$$

or

$$Mx_k + N\underset{\rightarrow k}{c} + Tc_\infty + V\underset{\rightarrow k+1}{r} \leq t \quad \Rightarrow \quad Mx_k + N\underset{\rightarrow k}{\tilde{c}} + Tc_\infty + Q\underset{\rightarrow k+1}{r} \leq t \tag{5.26}$$

At this point, the MCAS set can be described in a standard form as:

$$S_{MCASU} = \{x : \exists (\tilde{c}_k, c_\infty) \;\; s.t \; Mx_k + N\underset{\rightarrow k}{\tilde{c}} + Tc_\infty + Q\underset{\rightarrow k+1}{r} \leq t\} \tag{5.27}$$

where M, N, T and Q are suitable matrices and t is a vector of the limits. However, it should be noted that a standard admissible set algorithm may not terminate within a finite or reasonable time due to the implied steady-state being on a constraint boundary by virtue of $c_\infty \neq 0$ and thus some termination condition needs to be added.

Now, we can define the proposed algorithm which handles both constraints and advance information of the targets as follows.

Algorithm 5.1 *An OMPC algorithm with both advance knowledge handling and the potential to manage unreachable targets is summarised in the following optimisation:*

$$\min_{\underset{\rightarrow}{\tilde{c}},\ c_\infty} \quad W_1(c_\infty^T S c_\infty) + \underset{\rightarrow k}{\tilde{c}}^T S \underset{\rightarrow k}{\tilde{c}} \quad s.t. \quad Mx_k + N\tilde{c}_k + Tc_\infty + Qr_{k+1} \leq t \tag{5.28}$$

Use the optimised $\underset{\rightarrow}{\tilde{c}}, c_\infty$ *in conjunction with (4.24) to determine* c_k *and implement the first move* u_k *of the control law as defined in (5.1).*

5.6 Guarantees of feasibility and performance

This section establishes that Algorithm 5.1 has guarantees of recursive feasibility and asymptotic convergence to a point which minimises the weighted offset.

Lemma 5.3 *The proposed OMPC algorithm 5.1 with advance knowledge handling maintains feasibility irrespective of changes in the target.*

Proof: The proof follows in a straightforward fashion from the assumption of feasibility at start up and the inclusion of c_∞. If there were no change in the target, that is $r_{k+n_a+1} = r_{k+n_a}$, then one can use standard MPC arguments to show that the optimum (assumed feasible) solution from sample k can be carried forward to sample $k+1$ and thus feasibility is retained. In the case where $r_{k+n_a+1} \neq r_{k+n_a}$, one can always introduce a non-zero value of c_∞ such that the implied artificial target $\hat{r}_{k+n_a+1} = r_{k+n_a}$, thus again retaining feasibility. □

Remark 5.3 *It is not the purpose of this section to consider guarantees in the presence of disturbances as that case is far more demanding. The focus here is on a simple approach to dealing with the basic requirements. However, it is worth noting that the flexibility afforded in* c_∞ *is often sufficient to deal with any transient infeasibility caused by changes in disturbances.*

Theorem 5.1 *The proposed algorithm is convergent to the point, which minimises the weighted offset.*

Proof: At steady-state, the optimised values for c_k are all identical and therefore the optimisation is capped by:

$$\min_{c_k, c_\infty} J_p \quad \leq c_\infty^T [(W_1 + nI)S] c_\infty \tag{5.29}$$

Any optimised value of c_k such that $c_k^T S c_k < c_\infty^T S c_\infty$ would be a contradiction of the system being in steady-state and thus, noting the relationship of (5.4), the optimisation has minimised a weighted norm of the offset. □

5.7 Key observation

Now, it can be observed that Algorithm 5.1 can be applied for different scenarios as follows.

1. Unconstrained systems with no advance knowledge (fixed target). In this case, the degrees of freedom (d.o.f) are $\underset{\rightarrow k}{c}$ equal to zero, so the optimal control law is: ($u_k = -Kx_k$). In this case, $P_r \underset{\rightarrow k+1}{r}$ and $\underset{\rightarrow k}{\tilde{c}}$ and c_∞ must be equal to zero.

2. Unconstrained systems with advance knowledge (future target information). In this case, $\underset{\rightarrow k}{c}$ is active and the optimal control law becomes ($u_k = -Kx_k + c_k$). It shown in (4.17) that $\underset{\rightarrow k}{c} = P_r \underset{\rightarrow k+1}{r}$). In this case, $\underset{\rightarrow k}{\tilde{c}}$ and c_∞ must be equal to zero.

3. Constrained systems with advance knowledge (future target information). In this case $\underset{\rightarrow k}{\tilde{c}}$ is active due to constraints and the optimal control law is ($u_k = -Kx_k + P_r \underset{\rightarrow k+1}{r} + \tilde{c}_k$). In this case c_∞ must be equal to zero.

4. Constrained systems with advance knowledge (future target information) but with an unreachable target at steady-state. In this case the degrees of freedom $\underset{\rightarrow k}{\tilde{c}}$ c_∞ are active due to the constraints and the unreachable target, while the optimal control law is ($u_k = -Kx_k + P_r \underset{\rightarrow k+1}{r} + \hat{c}_k$) where, $\underset{\rightarrow k}{\hat{c}}$ includes both $\underset{\rightarrow k}{\tilde{c}}$ and c_∞.

5.8 Numerical examples for reachable/unreachable targets

This section gives examples, which demonstrate the efficacy of the proposed algorithm for handling both advance knowledge and unreachable targets for different systems such as stable, unstable, non-minimum phase, and MIMO systems, in a single simple optimisation. A common scenario is the infeasibility of the terminal constraints due to a change in the asymptotic target being too fast or too great. These examples demonstrate how the proposed algorithm smoothly introduces an artificial target during transients but moves to the correct steady-state asymptotically. The section also gives examples that show how easily the proposed algorithm can handle the permanent targets infeasibility (unreachable targets). Moreover, it is clear that the algorithm continues to embed information about how the target moves in a systematic fashion, thus improving performance compared to a more conventional approach with $n_a = 1$.

5.8.1 Target unreachable in transients but asymptotically reachable at steady-state

In this subsection, we will consider two examples of a target being unreachable in transients but being asymptotically reachable at steady-state, with advance knowledge. One is a SISO system and the other is a MIMO system. The advance knowledge n_a values are appropriately selected to be as close as possible to n_c to ease constraint handling, as discussed in the previous chapter.

Example 1: Two state dimensional SISO system

In this example, we consider a non-minimum phase, open-loop unstable model of the plant with dead time which has been adopted from [174]. The model matrices which describe the plant are given by:

$$A = \begin{bmatrix} 1.1053 & 0 \\ -0.01 & 0.8186 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0.0858 \end{bmatrix}; \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix} \qquad D = 0 \tag{5.30}$$

A closed-loop simulation for the system (5.30) is performed, with $n_c = 2$, $R = 0.1I$, $Q = C^T C$, subject to constraints on the input and the output as follows.

$$-5 \le u \le 1.8; \quad -1 \le y \le 2 \tag{5.31}$$

The advance knowledge is selected as $n_a = 4$ and the desired target is set to $r_k = 1$. This target is reachable in the steady-state, but not during transients.

Figure 5.1 shows the responses of the system (5.30) when the target is reachable in the steady-state, but not during transients, with the use of $n_a = 4$.

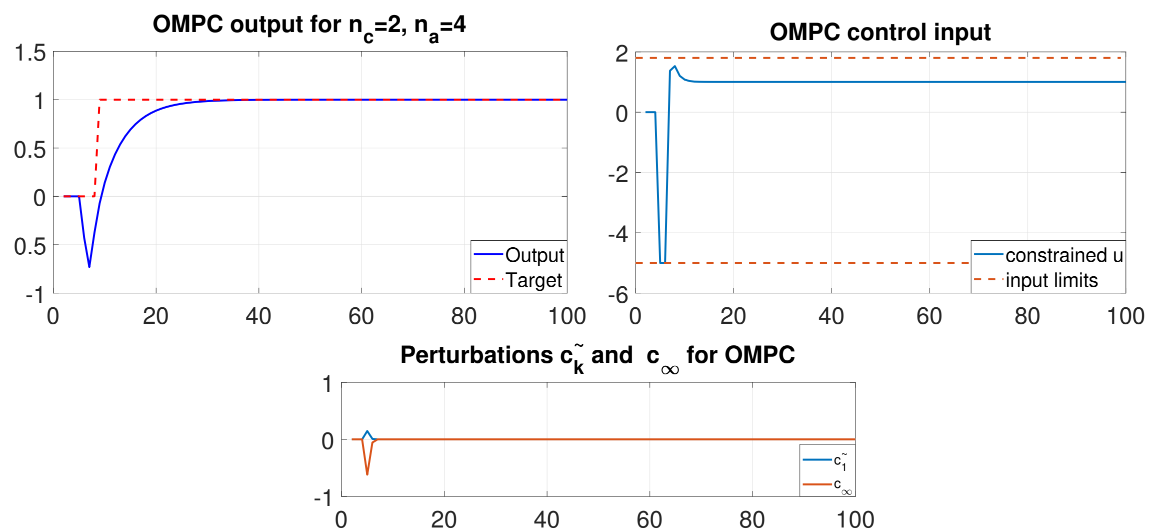

Figure 5.1: Closed-loop step responses for SISO system (5.30) for unreachable target during transients with $n_a = 4$.

It is clear from Figure 5.1 (upper left plot), that Algorithm 5.1 provides effective control for a constrained system with advance knowledge. Readers will note that the term c_∞ (lower plot) is non-zero during the transients and zero for a longer time, as expected, since the target is infeasible during transients. It is also noted that the term $\tilde{c}_k$ (lower plot) is non-zero during the transients and zero for a longer time, as expected, but this is because of the input saturation (upper right plot) in the transients. This indicates that the target asymptotically becomes feasible (reachable) at steady-state. It will be also noted that the input perturbation c_k is non-zero for longer, as expected from equation (5.6).

Example 2 : Four state dimensional MIMO system

In this example, we will consider the following four dimensional MIMO system which was introduced in [165].

$$A = \begin{bmatrix} 0.9146 & 0.0 & 0.0405 & 0.1 \\ 0.1665 & 0.1353 & 0.0058 & -0.2 \\ 0.0 & 0.0 & 0.1353 & 0.5 \\ -0.2 & 0 & 0 & 0.8 \end{bmatrix}, B = \begin{bmatrix} 0.0544 & -0.0757 \\ 0.0053 & 0.1477 \\ 0.8647 & 0.0 \\ 0.5 & 0.2 \end{bmatrix}, \qquad (5.32)$$

$$C = \begin{bmatrix} 1.7993 & 13.216 & 0.0 & 0.0 \\ 0.8233 & 0.0 & 0.0 & -0.3 \end{bmatrix}, D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

.

We perform a closed-loop simulation for the system (5.32), with $n_c = 2,\ R = diag(0.01, 0.01)$ and $Q{=}C^TC$, subject to constraints on the inputs and outputs as follows.

$$\begin{bmatrix} -1 \\ -2 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 2 \end{bmatrix}; \qquad \begin{bmatrix} 1.7993 & 13.2160 & 0 & 0 \\ 0.8233 & 0 & 0 & -0.3 \\ -1.7993 & -13.2160 & 0 & 0 \\ -0.8233 & 0 & 0 & 0.3 \end{bmatrix} x_k \leq \begin{bmatrix} 7 \\ 1 \\ 3 \\ 1 \end{bmatrix} \qquad (5.33)$$

The advance knowledge is selected to be $n_a = 3$. The desired target is set to $r_1 = 1$ and $r_2 = 0.3$. These targets are reachable in the steady-state, but not during transients.

Figure 5.2 shows the closed-loop step responses for the MIMO system (5.32) when the targets are reachable in the steady-state, but not during transients with the use of $n_a = 3$.

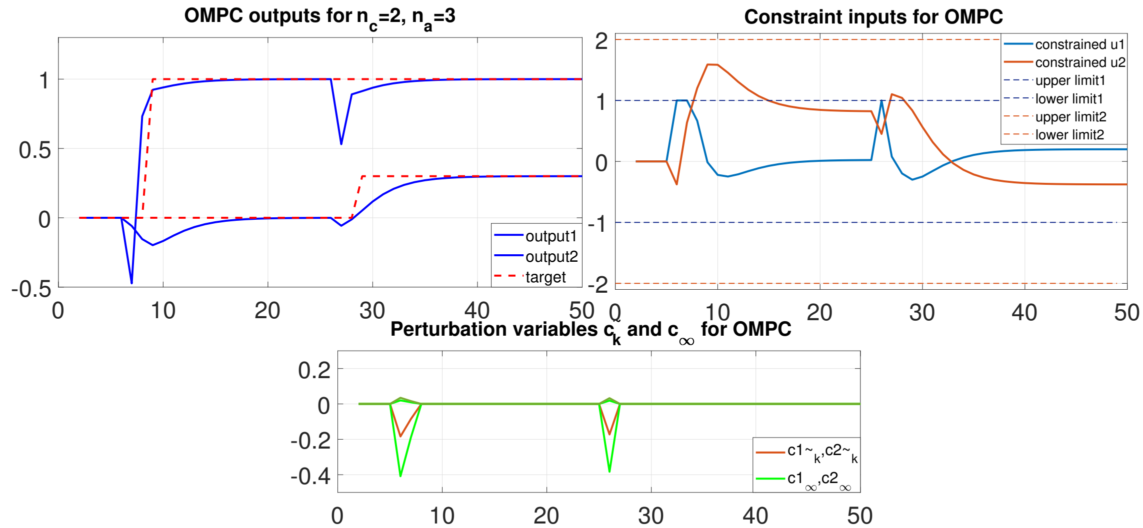

Figure 5.2: Closed-loop step responses for MIMO system (5.32) for unreachable target targets during transients with $n_a = 3$.

Again, it is clear from Figure 5.2 that Algorithm 5.1 provides effective control for constrained MIMO systems with advance knowledge. Both output1 and output2 (upper left plot) asymptotically follow the associated targets, $r_1 = 0.3$ and $r_2 = 1$. It will be noted that the term $\tilde{c}_k$ (lower plot) is non-zero during the transients and zero for a longer time, as expected. This is because the first control input (upper right plot) is active during the transients only. It will be also noted that the term c_∞ (lower plot) is non-zero during the transients, as expected, but tends to be zero for a longer time since the target is reachable at steady-state. The input perturbation, c_k is non-zero for longer, as expected from equation (5.6) and discussed in the previous example.

5.8.2 Target unreachable during both transient and steady-state

In this subsection, we will consider two example scenarios where the target unreachable in both the transient and steady-state, with advance knowledge. One example is for a SISO system and the other for a MIMO system. The advance knowledge n_a values are appropriately selected to be as close as possible to n_c to ease constraint handling, as discussed in the previous chapter.

Example 3: Two state dimensional SISO system

In this example, we will consider the following non-minimum phase model, introduced in [84] whose system matrices are given by:

$$A = \begin{bmatrix} 0.22 & 0.44 \\ 0.0 & 0.88 \end{bmatrix}, B = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}; \quad C = \begin{bmatrix} 1.0 & 0.0 \end{bmatrix}, D = \begin{bmatrix} 0 \end{bmatrix} \tag{5.34}$$

We perform closed-loop simulation for the system (5.34), with $n_c = 3$, $R = I$, $Q = C^T C$ and advance knowledge is selected as $n_a = 5$. The desired target is set to $r = 4$, which is unreachable during both the transients and steady-state. The system is subject to constraints on the input and output as follows.

$$\begin{bmatrix} -1 \end{bmatrix} \leq \begin{bmatrix} u \end{bmatrix} \leq \begin{bmatrix} 0.65 \end{bmatrix}; \qquad \begin{bmatrix} -6 \end{bmatrix} \leq \begin{bmatrix} y \end{bmatrix} \leq \begin{bmatrix} 6 \end{bmatrix} \tag{5.35}$$

Figure 5.3 shows the responses of the system (5.34) when the target is unreachable ($r = 4$) in the steady-state with the use of $n_a = 5$.

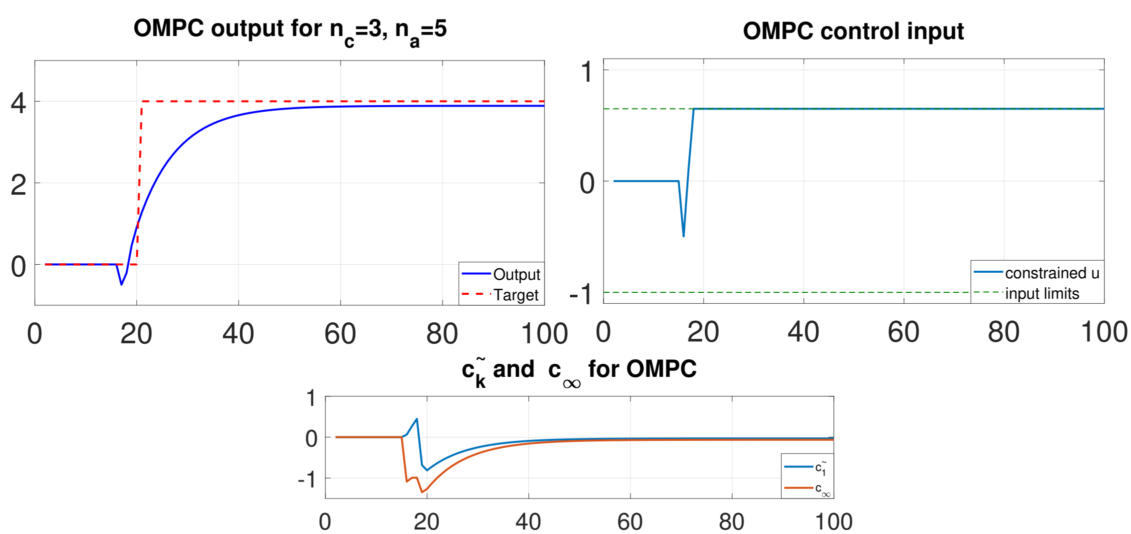

Figure 5.3: Closed-loop step responses for system (5.34) for unreachable target with $n_a = 5$.

For the system of (5.34), the asymptotic targets are unreachable (upper left plot) as this would require a violation of the input constraint. Nevertheless, the proposed algorithm has handled the unreachable target in a sensible fashion and gives good performance which takes the output as close as possible to the desired target while maintaining good behaviour, and effective anticipation, during transients. It will be noted that the term c_∞ (lower plot) is non-zero during both the transients and steady-state, as expected since the target is infeasible during both transients and steady-states. It is also noted that the term $\tilde{c}$ (lower plot) is non-zero for the long term since the input constraints (upper right) are active for that time.

Example 4: Four state dimensional MIMO system

In this example, we consider the model of the longitudinal motion of aircraft, studied in [19]. The matrices which describes the model are given by:

$$A = \begin{bmatrix} 0.9996 & 0.0383 & 0.0131 & -0.0322 \\ -0.0056 & 0.9647 & 0.7446 & 0.0001 \\ 0.0020 & -0.0097 & 0.9543 & 0 \\ 0.0001 & -0.0005 & 0.0978 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0001 & 0.1002 \\ -0.0615 & 0.0183 \\ -0.1133 & 0.0586 \\ -0.0057 & 0.0029 \end{bmatrix}, \tag{5.36}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 7.74 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

We perform closed-loop simulation for the system (5.36) with advance knowledge $n_a = 5$. The tuning parameters can be set as follows. The control horizon, $n_c = 2$, the weighting matrix, $Q = C^T C$ and the weight in the input command, $R = diag(0.01, 0.01)$ subject to constraints on the inputs and outputs, as follows.

$$\begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}; \qquad \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix} \leq \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \leq \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} \tag{5.37}$$

The desired target is set to $r_1 = 0.5$ and $r_2 = 1.6$. These targets are unreachable during both the transients and steady-state since the desired target $r_1 = 0.5$ is outside the output limits as shown in (5.37).

Figure 5.4 shows the closed-loop step responses for the MIMO system (5.36) when the targets are unreachable ($r_1 = 0.5$ and $r_2 = 1.6$) in steady-state with the use of $n_a = 5$.

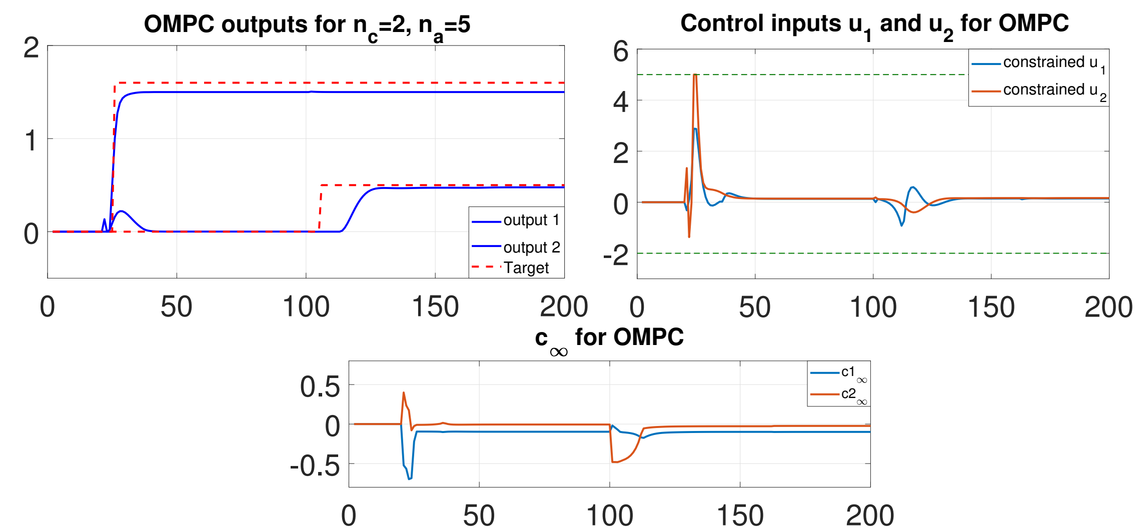

Figure 5.4: Closed-loop step responses for the system of (5.36) for unreachable targets with $n_a = 5$.

For the system of (5.36), the asymptotic targets are unreachable, as this will require a violation of the input constraint. Nevertheless, the proposed algorithm has handled the unreachable target in a sensible fashion and gives good performance which takes the output as close as possible to the desired target (upper left plot) while maintaining good behaviour, and effective anticipation, during transients. It will be noted that the terms $c_{\infty 1}$ and $c_{\infty 2}$ (lower plot) remain non-zero during both the transients and steady-state, as expected. The term $\tilde{c_k}$ is also non-zero during the transients since the input for the second output (upper right plot) is saturated (with active constraints) at the times of the set point changes; hence, the term c_k is non-zero for longer, as expected from (5.6).

In order to demonstrate the benefits of using advance knowledge, we will perform a comparison between the performance indices of the corresponding systems with and without advance knowledge for both reachable and unreachable targets, in tabular form as shown in Table 5.1.

It is clear from Table 5.1 that the performance indices of the corresponding systems with advance knowledge are lower than those without advance knowledge. This indicates the advantage of including advance knowledge in the corresponding feed-forward design.

Examples/cost functions	J with $n_a = 1$	J with $n_a = 5$
Examples for system (5.30)	42.15	32.08
Examples for system (5.32)	2.5	0.80
Examples for system (5.34)	35.91	25.79
Examples for system (5.36)	188.18	168.57

Table 5.1: Performance indices for step changes in the target for systems (5.30, 5.32, 5.34, and 5.36)

5.9 Conclusion

At times, the desired target will be unreachable and, in such scenarios, a default MPC algorithm becomes ill-defined This chapter has made a contribution by showing how a simple framework can be implemented to incorporate unreachable targets thus, giving a transparent view of how the control choices deviate from the ideal, due to both advance knowledge and infeasible targets. The proposed framework is simple and caters for both transient and permanent infeasibility in the target without the need to change the algorithm on-line. The algorithm has also shown how the systematic embedding of advance information is straightforward and beneficial. Moreover, the proposed framework provides insights for designers and also offers simple handling for trading off transient performance against convergence to the asymptotic steady-state. Further illustration is presented in Chapter 8.

Chapter 6

EFFICIENT ROBUST MPC TRACKING FOR UNCERTAIN SYSTEMS

6.1 Introduction

Much of the existing literature on robust MPC uses LMIs and/or tube based techniques. Both of these have weaknesses due to either the high computational complexity, meaning that extension to the higher dimensions is challenging, or simply because the set definitions are conservative. Moreover, of specific importance here is that the extension to full tracking scenarios utilising significant advance knowledge is not obvious [8].

One technique that seems to have greater potential for extension is based on the maximal admissible sets (MAS) [52, 114]. It was already shown in [113] that this provides a simple robust MPC approach for the regulation case and thus the expectation is that extension to the tracking case with significant advance knowledge should be relatively straightforward and critically, produce a computationally efficient algorithm. Therefore, this chapter aims to design a simple and efficient robust tracking MPC algorithm for linear parameter varying (LPV) system subject to state and input constraints and with significant advance knowledge included systematically. The approach is an extension of [113] to tracking scenarios, thus incorporating the parametric uncertainty in the optimisation problem and also using some incorporating concepts from [158, 86] to deal with unreachable targets. The main contributions are:

1. Systematic incorporation of future target information into a robust tracking MPC

optimisation problem.

2. Definition and construction of an appropriate robust invariant set which includes significant advance information on the target.

3. Incorporating flexibility to deal with unreachable targets.

The chapter is organised as follows. Section 6.2 presents a discussion on how the MCAS can be defined for uncertain systems. Section 6.3 shows how to derive an MCAS for the regulator problem for uncertain systems. Section 6.4 discusses robust tracking MPC where the target is reachable, while Section 6.5 discusses it for scenarios where the target is unreachable. Section 6.6 summarises the robustness to uncertainty while Section 6.7 presents numerical examples. The chapter finishes by offering some conclusions in Section 6.8.

6.2 Generic MCAS for uncertain systems

In this section, we will explore the background to MCAS construction for uncertain systems in order to study the design of a robust MPC tracking algorithm using the concept of MCAS.

6.2.1 LPV system model

In this chapter, we consider the discrete time LPV in full system

$$x_{k+1} = A(k)x_k + B(k)u_k. \tag{6.1}$$

$A(k), B(k)$ are the matrices defining the model.

Parameter uncertainty is quantified by $[A(k)\, B(k)] \in \Omega = Co\,\{[A_1\; B_1], ..., [A_m\; B_m]\}$, where Co refers to the convex hull of the extreme models, in which $[A\; B] \in \Omega$; hence: $0 \le \lambda_i \le 1,\; \sum \lambda_i = 1$ and $[A\; B] = \sum_{i=1}^{L} \lambda_i [A_i\; B_i]$.

Let us assume that the system with input and state constraints can be expressed in inequalities as:

$$\underline{u} \le u_k \le \bar{u}, \qquad K_{xmax} x_k \le \bar{x}, \qquad k = 1, \cdots, \infty \tag{6.2}$$

6.2.2 The state feedback controller K

It is common in robust MPC control approaches [105, 6], for the state feedback controller K to be determined on-line so that K varies for every sample. However, the aim here is to build on the approach of [113] which utilises a fixed K and combines this with the ideas summarised in the previous section. This has the advantage of keeping the overall algorithm complexity similar to the nominal case and indeed the on-line computational load differs only because the implied number of linear inequalities increases.

Nevertheless, a key point mentioned in [114] is that there must exist an invariant set for the uncertain unconstrained closed-loop dynamics, $x_{k+1} = \Phi(k)x_k$ with fixed K. Consequently, it is important to check the quadratic stability of the underlying closed-loop system. This can be achieved by satisfying the stability condition using a linear matrix inequality (LMI) technique.

$$\exists P = P^T > 0 \quad s.t. \;\; \Phi_i^T P \Phi_i \leq P, \qquad i = 1, ..., m. \tag{6.3}$$

where $\Phi_i = A_i - B_i K$ is a transition matrix for the LPV model.

Remark 6.1 *Algorithms for identifying a K to satisfy (6.3) and simultaneously optimise a nominal cost function are readily available in the literature on LMI techniques, e.g. [71]. However, one could argue that the corresponding K for the nominal case may also be preferred where it satisfies (6.3).*

6.2.3 Derivation of robust MCAS

The MCAS for time-varying targets was introduced in Section 4.5.2. It was shown that this MCAS can be deployed for systems in which the model predictions are well-defined if they satisfy the constraints for all future samples. This prediction model is given by:

$$Z_{k+1} = \Psi Z_k \tag{6.4}$$

where Ψ is a transition matrix and Z_k is an augmented state of the autonomous model.

It is also shown in Section 4.5.2 that the input and state constraints can be expressed as one inequality of sample constraints as follows.

$$GZ_k \leq f, \qquad \forall k \tag{6.5}$$

This sample constraints inequality is split into two parts. One part (say $G_1 Z_k \leq f_1$) is time-varying and is checked for each sample k while the other part (say $G_2 Z_k \leq f_2$) is fixed and so checked only for $k = 0$.

Thus, it is convenient to describe sample constraint (6.5) as:

$$\begin{bmatrix} G_1 \\ --- \\ G_2 \end{bmatrix} Z_k \leq \begin{bmatrix} f_1 \\ --- \\ f_2 \end{bmatrix}, \quad \forall k \tag{6.6}$$

Because G_2 is fixed, their rows will not be carried forward in any admissible iteration.

Next, we wish to modify MCAS equation (2.96), which was described in Section 4.5.2, in order to utilise the structure in equation (6.6). Specifically, the iteration only needs to take place on the rows in G_1 and thus excludes G_2.

Hence, the structure of the admissible set can be given for a specific horizon 'n' as:

$$\underbrace{\begin{bmatrix} G_2 \\ --- \\ G_1 \\ G_1\Psi^1 \\ G_1\Psi^2 \\ \vdots \\ \vdots \end{bmatrix}}_{F} Z_k \leq \underbrace{\begin{bmatrix} f_2 \\ --- \\ f_1 \\ f_1 \\ f_1 \\ \vdots \\ \vdots \end{bmatrix}}_{t} \tag{6.7}$$

Thus, the MCAS can be defined in a standard form as:

$$FZ_k \leq t \tag{6.8}$$

However, the MCAS discussed above is useful only for nominal cases. In this section, we will demonstrate how this MCAS can be extended to be implemented on uncertain (LPV) systems.

The key point is that the closed-loop uncertain system matrices A and B are no longer fixed (unknown). This implies that the transition matrix is not fixed and is denoted as $\Psi(k)$.

Thus, the prediction model can be defined in terms of an autonomous model as:

$$Z_{k+1} = \Psi(k) Z_k \tag{6.9}$$

where $\Psi(k) \in Co\{\Psi_1, ...\Psi_m\}$.

Building on the prediction model of (6.9) and the sample constraints of (6.6), one can describe the structure of the admissible set in a standard form as:

$$\underbrace{\begin{bmatrix} G_2 \\ --- \\ G_1 \\ -- \\ G_1\Psi_1 \\ \vdots \\ G_1\Psi_m \\ -- \\ G_1\Psi_1\Psi_1 \\ \vdots \\ G_1\Psi_1\Psi_m \\ --- \\ \vdots \end{bmatrix}}_{G_r} Z_k \leq \underbrace{\begin{bmatrix} f_2 \\ --- \\ f_1 \\ f_1 \\ f_1 \\ \vdots \\ f_{1,m} \end{bmatrix}}_{f_r} \tag{6.10}$$

As shown, structure (6.10) is intractable in general. Therefore, we need an alternative approach to define a simple structure in order to determine the MCAS.

An appropriate approach may be to add one inequality at a time and check the redundant constraints. We will use this approach, but not in detail, since they are already published in [114]. In fact, this is illustrated in Algorithm 2.1.

Algorithm 6.1 *We assume that the transition matrix* $\Psi(k)$ *has stable properties. This implies that* $\lim_{k\to\infty} \Psi^k = 0$.

Building on the structure of (6.10), the robust MCAS can be determined as follows:

1. *For i=1,...,m, compute the transition matrices* Ψ_i.

2. *Generate* G_1, G_2 *and* f_1, f_2.

3. *Implement Algorithm 2.1 in Chapter 2 to determine the matrix* F_r *and limits vector* d_r

Summary: If the input and state constraints are described as $G_1 Z_k \leq f_1$ and $G_2 Z_k \leq f_2$ and the dynamics of the uncertain system are described as $Z_{k+1} = \Psi_i Z_k, \quad (i = 0, \ldots, m)$, then we can easily obtain the MCAS.

6.3 Robust MCAS for the regulation case

In the previous section we showed how to construct a generic MCAS for uncertain systems. Now, we will show how these results can be used to construct an MCAS for the regulation case. In this case, the model predictions (6.9) can be described as:

$$Z_{k+1} = \begin{bmatrix} \Phi_i & [B_i, 0..., 0] \\ 0 & D_C \end{bmatrix} \begin{bmatrix} x_k \\ \underrightarrow{c}_k \end{bmatrix} \tag{6.11}$$

where

$$D_c = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{6.12}$$

Now, it is convenient to describe the input and state constraints (6.2) in terms of the state Z_k as follows.

$$\begin{bmatrix} -K & [I, 0, 0, ...] \\ K & -[I, 0, 0, ...] \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{u} \\ \underline{u} \end{bmatrix}; \quad \begin{bmatrix} K_{xmax} & 0 \\ -K_{xmax} & 0 \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{x} \\ \underline{x} \end{bmatrix} \tag{6.13}$$

Equation(6.13) can be defined as:

$$G_1 Z_k \leq f_1 \tag{6.14}$$

where G_1 describes the sample constraints and f_1 describes the state and input limits.

Remark 6.2 *For the regulation case, G_2 is not needed as the origin is assumed to be feasible.*

It is noted that the components G_1 and f_1 are fixed, as expected. This is obvious because the feedback controller K is fixed. The only difference between the nominal and uncertain cases is the transition matrix Ψ, as shown in the summary of the previous section.

Building on the transition matrix in (6.11) and inequality (6.14), and using the admissible set of Algorithm 2.1, we determine the MCAS for the regulation case with an uncertain system.

Thus, the robust MCAS for the regulation case is defined in compact form as:

$$F_r Z_k \leq d_r \tag{6.15}$$

or in expanded form as:

$$S_{RMCAS} = \left\{ x : \exists \underset{\rightarrow k}{c} \quad s.t \quad M_r x_k + N_r \underset{\rightarrow k}{c} \leq d_r \right\} \tag{6.16}$$

where M_r, N_r are suitable matrices and d_r is a vector of the limits.

6.4 Robust tracking MPC for reachable targets

Tracking MPC scenarios were discussed earlier, in Chapter 4, where they account for the changing targets and incorporation of advance knowledge, but with nominal dynamics only. In this section, we extend this approach to be applicable to systems with a poly-topic uncertainty model while retaining guarantees of recursive feasibility and convergence. This section will discuss robust MPC tracking for reachable targets.

Subsection 6.4.1 re-parameterises the input predictions and defines a robust cost function, while Subsection 6.4.2 presents a definition of the uncertain model in a simple form. Subsection 6.4.3 defines the MCAS for uncertain systems, assuming a reachable target. Subsection 6.4.4, the final subsection, introduces a proposed algorithm for robust OMPC tracking.

6.4.1 Input parametrisation and cost function for reachable targets

As discussed in Section 4.1, equation (4.4), the state and input predictions, which are suitable for time-varying targets with the nominal case, are given by:

$$\begin{cases} x_{k+1} - x_{ss|k} = A(x_k - x_{ss|k}) + B(u_k - u_{ss|k}); \quad u_k - u_{ss|k} = -K(x_k - x_{ss|k}) + c_k, \quad k < n_c \\ x_{k+1} - x_{ss|k} = A(x_k - x_{ss|k}) + B(u_k - u_{ss|k}); \quad u_k - u_{ss|k} = -K(x_k - x_{ss|k}), \quad k \geq n_c \end{cases} \tag{6.17}$$

Similarly, the predictions for the uncertain systems can be described simply by replacing A and B in equation (6.17) by $A(k)$ and $B(k)$, respectively since they are no longer fixed in this case. At this point, one can rewrite the state predictions as:

$$x_{k+1} - x_{ss|k} = A(k)(x_k - x_{ss|k}) + B(k)(u_k - u_{ss|k}) \quad . \tag{6.18}$$

and the input predictions as:

$$\begin{cases} u_k - u_{ss|k} = -K(x_k - x_{ss|k}) + c_k, \quad k < n_c \\ u_k - u_{ss|k} = -K(x_k - x_{ss|k}), \quad k \geq n_c \end{cases} \tag{6.19}$$

The nominal cost function which includes future target information is well defined in Chapter 4, equation (4.16), as:

$$\arg \min_{\underset{\rightarrow k}{c}} J \equiv \arg \min_{\underset{\rightarrow k}{c}} \{ \underset{\rightarrow k}{c}^T S_c \underset{\rightarrow k}{c} + 2 \underset{\rightarrow k}{c}^T S_{cr} \underset{\rightarrow k+1}{r} \} \tag{6.20}$$

It has been shown that using this nominal function will focus on the perturbation c_k rather than the 'worst case' J. However, the worst case requires a min-max approach so that it is non-trivial.

In order to avoid complexity for the robust case, we will use a performance index, which is based on a control perturbation to the loop. It might not lead to the optimum robust performance for uncertain systems, but it is at least a pragmatic and simple in nature.

The following section shows the derivation of the cost function which can be used for uncertain cases.

As discussed in Theorem 4.1 in Chapter 4, the minimisation of performance index (6.20) gives the same optimum $\underset{\to k}{c}$ as the following optimisation.

$$\underset{\to}{\tilde{c}} = \arg\min_{\underset{\to}{\tilde{c}}} \quad J = \underset{\to k}{\tilde{c}}^T S_c \underset{\to k}{\tilde{c}} \tag{6.21}$$

Remark 6.3 *The reader may wonder whether optimising the performance index (6.21) leads to closed-loop stability. However, it is easy to demonstrate that this is the case, assuming that we can guarantee recursive feasibility. The use of the MCAS (6.16) implies the feasibility of the tail which in turn implies the monotonicity of J. If a feedback control loop in Figure 6.1, is robust stable and $\tilde{c}$ is convergent, hence the feedback control loop must be stable.*

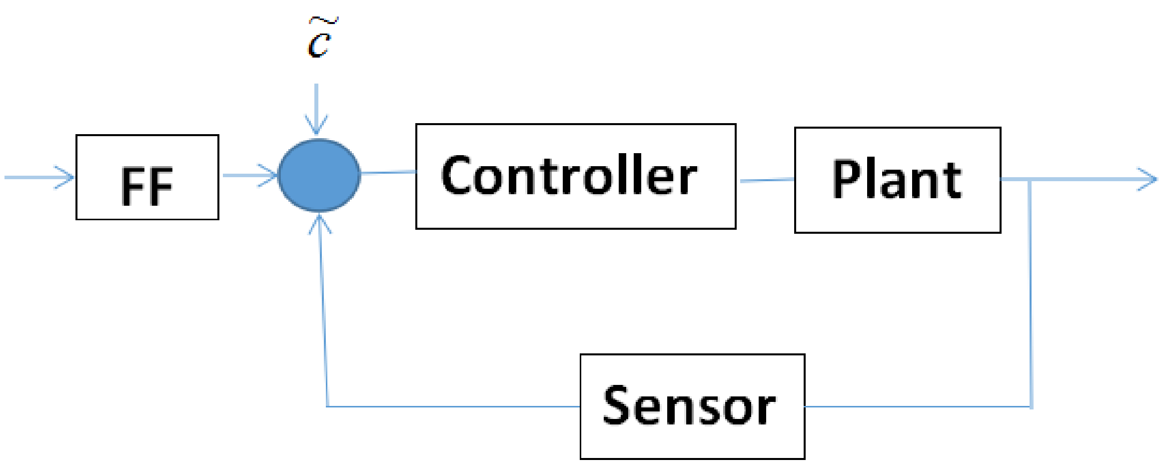

Figure 6.1: Closed-loop control for uncertain systems with advance knowledge

.

According to the above discussion, the cost function J (6.21) remains valid for the robust case, so we can use the robust MCAS as part of optimising J such that the control perturbation $\tilde{c}$ converges.

6.4.2 The autonomous model for uncertain system predictions

It is important to capture the variability in the closed-loop trajectories due to the uncertainty within the model parameters. One can capture this uncertainty efficiently using a set of linear inequalities [114], provided that the dynamics and constraints can be captured in appropriate form, as shown in equations (6.4, 6.5).

The basic algorithm requires a one-step ahead state evolution equation (analogous to (5.11)) and a statement of constraint dependence on the state at each sampling instant.

Thus, the dual-mode predictions for system (6.1) with control law (6.18) can be described as:

$$\begin{array}{rcl} x_{k+1+i} & = & \Phi(k)x_{k+i} + (I-\Phi(k))K_{xr}(r_{k+1+i}) + B(k)c_k \\ u_{k+i} & = & -Kx_{k+i} + [KK_{xr} + K_{ur}](r_{k+1+i}) + c_k \end{array} \tag{6.22}$$

where $\Phi(k) = A(k) - B(k)K$. It is noted that

$$[A(k)\ B(k)] \in \Omega \quad \Rightarrow \quad \Phi(k) \in Co\,\{\Phi_1, ..., \Phi_m\}\,.$$

Remark 6.4 *The parameters K_{ur} and K_{xr} are fixed since they and control law (6.19) are defined in terms of the nominal system matrices A and B.*

Now, it is convenient to substitute the perturbation of equation (4.24) into equation (6.22); thus, one can rewrite the closed-loop dynamics in terms of $\underset{\rightarrow k}{\tilde{c}}$ as:

$$x_{k+1+i} = \Phi(k)x_{k+i} + [B_i, \cdots, 0]\underset{\rightarrow k}{\tilde{c}} + [[B_i, \cdots, 0]Pr + [(I-\Phi(k))K_{xr}, \cdots, 0]]\underset{\rightarrow k+1+i}{r} \tag{6.23}$$

The control input is given by:

$$u_{k+i} = -Kx_{k+i} + [I, \cdots, 0]\underset{\rightarrow k}{\tilde{c}} + [[I, \cdots, 0]Pr + [KK_{xr} + K_{ur}, \cdots, 0]]\underset{\rightarrow k+1+i}{r} \tag{6.24}$$

Lemma 6.1 *The uncertain dual-mode system predictions of (6.23) can be captured in a single mode autonomous model of the following form:*

$$Z_{k+1} = \Psi(k)Z_k;\ \ Z_k = [x_k^T, \underset{\rightarrow k}{\tilde{c}}^T, \underset{\rightarrow k+1}{r}^T]^T \tag{6.25}$$

$$\Psi(k) = \begin{bmatrix} \Phi(k) & [B(k), 0..., 0] & [B(k), 0, ..., 0]Pr + [(I - \Phi(k))K_{xr}, 0, .., 0] \\ 0 & D_c & 0 \\ 0 & 0 & D_R \end{bmatrix} \tag{6.26}$$

where $\Psi(k) \in Co\{\Psi_1, ..., \Psi_m\}$ and D_C, D_R are shift matrices, as defined in (4.5) and (4.6).

Lemma 6.2 *The quadratic invariance of the closed-loop dynamic $\Phi(k)$ is sufficient to ensure the quadratic invariance of the augmented dynamic $\Psi(k)$.*

Proof: The additional dynamics in $\Psi(k)$, as compared to $\Phi(k)$, relate to the variables $\tilde{c}_k$, $\underrightarrow{r}_{k+1}$. These dynamics are governed solely by shift matrices and thus must converge to fixed, possibly non-zero, values. □

Summary: Given the model predictions of uncertain system (6.22) and adding a perturbation about the nominal, then one can define the uncertain model in a simple form as: $Z_{k+1} = \Psi(k)Z_k, \qquad (k = 0, \ldots, m)$ and $Z_k = [x_k^T, \underrightarrow{\tilde{c}}_k^T, \underrightarrow{r}_{k+1}^T]^T$.

6.4.3 Derivation of robust MCAS for a reachable target

In the previous section, we have discussed how to define the MCAS for uncertain systems for the regulation case. In this section, we will demonstrate how this MCAS can be extended to deal with tracking scenarios. We consider the transition matrix (6.26). As discussed in the previous section, to ensure convergence, input and state constraints can be combined and described as an inequality in terms of the augmented state variable in (6.25), as:

$$\begin{bmatrix} G_1 \\ --- \\ G_2 \end{bmatrix} Z_k \leq \begin{bmatrix} f_1 \\ --- \\ f_2 \end{bmatrix}, \quad \forall k \tag{6.27}$$

Following the steps outlined in the previous section, one can derive the corresponding G_1, G_2, f_1 *and* f_2, for each expected model as follows.

The constraints can be classified as two categories. One category describes the transient constraints and the other describes the steady-state constraints. At this point, we will describe those constraints in terms of the augmented state as follows.

Transient constraints

Again, the input and state constraints of (6.2) are classified as transient constraints, since they are active at the transients. These constraints can be expressed in terms of the augmented state Z_k of (6.25) as:

$$\begin{bmatrix} -K & [I,0,0,...] & [1,0,...,0]Pr + [KK_{xr}+K_{ur},0,..,0] \\ K & -[I,0,0,...] & -[1,0,...,0]Pr - [KK_{xr}+K_{ur},0,..,0] \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{u} \\ \underline{u} \end{bmatrix} \tag{6.28}$$

$$\begin{bmatrix} K_{xmax} & 0 & 0 \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{x} \end{bmatrix} \tag{6.29}$$

The steady state and input constraints at transient, x_{ss} and u_{ss}, respectively, are also classified as transient constraints and can be defined as:

$$\lambda \underline{u} \leq K_{ur} r_{k+1} \leq \lambda \bar{u} \tag{6.30}$$

$$K_{xmax} K_{xr} r_{k+1} \leq \lambda \bar{x} \tag{6.31}$$

Similarly, the constraints of (6.30) and (6.31) can be expressed in the form:

$$\begin{bmatrix} 0 & 0 & [K_{ur},0,\cdots,0] \\ 0 & 0 & [-K_{ur},0,...,0] \end{bmatrix} Z_k \leq \begin{bmatrix} \lambda\bar{u} \\ \lambda\underline{u} \end{bmatrix} \tag{6.32}$$

$$\begin{bmatrix} 0 & 0 & [K_{xr},0,\cdots,0] \end{bmatrix} Z_k \leq \begin{bmatrix} \lambda\bar{x} \end{bmatrix} \tag{6.33}$$

Now, it is straightforward to define the elements of G_1, by combining the sets (6.28, 6.29, 6.32, 6.33) together, giving that:

$$\underbrace{\begin{bmatrix} -K & [I,0,0,...] & [1,0,...,0]Pr + [KK_{xr}+K_{ur},0,..,0] \\ K & -[I,0,0,...] & -[1,0,...,0]Pr + [KK_{xr}+K_{ur},0,..,0] \\ 0 & 0 & [K_{ur},0,\cdots,0] \\ 0 & 0 & [-K_{ur},0,...,0] \\ 0 & 0 & [K_{xr},0,\cdots,0] \\ K_{xmax} & 0 & 0 \end{bmatrix}}_{G_1} \underbrace{\begin{bmatrix} x_k \\ \underset{\rightarrow}{c}_k \\ \underset{\rightarrow}{r}_{k+1} \end{bmatrix}}_{Z_k} \leq \underbrace{\begin{bmatrix} \bar{u} \\ \underline{u} \\ \lambda\bar{u} \\ \lambda\underline{u} \\ \lambda\bar{x} \\ \bar{x} \end{bmatrix}}_{f_1}, \quad \forall k. \tag{6.34}$$

These elements need to be tested for every sample. It is noted that G_1 is fixed since the associated elements are fixed.

Steady state constraints

In the previous subsection, we showed how to derive G_1. Now will show how to derive G_2, which captures the steady-state constraints.

For uncertain systems, the actual steady-state constraints are uncertain, so we try to capture all possible steady-states for the uncertain model.

First, we need to define u_{ssi} and x_{ssi}

Analogous to equation (2.9) in Chapter 2, the steady-state and input, x_{ssi}, u_{ssi}, respectively can be computed as:

$$\begin{bmatrix} C & 0 \\ A_i - I & B_i \end{bmatrix} \begin{bmatrix} r \\ 0 \end{bmatrix} = \begin{bmatrix} x_{ssi} \\ u_{ssi} \end{bmatrix}, \qquad i = 1, \cdots\cdots, m \tag{6.35}$$

Thus, the solution of these steady-states can be given by:

$$\begin{bmatrix} x_{ssi} \\ u_{ssi} \end{bmatrix} = \begin{bmatrix} K_{xri} \\ K_{uri} \end{bmatrix} r_{k+na}, \qquad i = 1, \cdots\cdots, m \tag{6.36}$$

where K_{uri} and K_{xri} are suitable parameters.

In this case, the steady-state (u_{ssi}, x_{ssi}) is affected by model uncertainty.

Hence, as discussed in Section 5.5, the steady-state constraints can be expressed in an inequalities form as:

$$\lambda \underline{u} \leq K_{uri} r_{k+na} \leq \lambda \bar{u}, \qquad i = 1, \cdots\cdots, m \tag{6.37}$$

$$\lambda \underline{x} \leq K_{xri} r_{k+na} \leq \lambda \bar{x}, \qquad i = 1, \cdots\cdots, m \tag{6.38}$$

Since the model matrices $[A(k)\ B(k)] \in \Omega$, one can define the constraints in terms of the corresponding matrices for each A_i, B_i. Then we can define steady-state values in terms of K_{uri} and K_{xri} using equation (6.36).

Thus, we can express the steady-state and input constraints (6.37) and (6.38), respectively, in terms of the autonomous model as:

$$\begin{bmatrix} 0 & 0 & [0,\cdots,0,K_{uri}] \\ 0 & 0 & [0,.,.,.,0,-K_{uri}] \\ 0 & 0 & [0,\cdots,0,K_{xri}] \end{bmatrix} Z_k \leq \begin{bmatrix} \lambda\bar{u} \\ \lambda\underline{u} \\ \lambda\bar{x} \end{bmatrix}, \quad i = 1,\cdots\cdots,m \tag{6.39}$$

To ensure a feasible solution, it is necessary to set up a sensible target. Therefore, it is convenient to express the allowable range of the target by its upper and lower limits as:

$$\begin{bmatrix} 0 & 0 & I \\ 0 & 0 & -I \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{r} \\ \underline{r} \end{bmatrix} \tag{6.40}$$

Now, we can define the elements of G_2 by combining sets (6.39) and (6.40) together, given that

$$\underbrace{\begin{bmatrix} 0 & 0 & [0,..0,K_{ur1}] \\ \vdots & \vdots & \vdots \\ 0 & 0 & [0,..0,K_{urm}] \\ 0 & 0 & [0,..,0,-K_{ur1}] \\ \vdots & \vdots & \vdots \\ 0 & 0 & [0,..,0,-K_{urm}] \\ 0 & 0 & [0,..,0,K_{xr1}] \\ \vdots & \vdots & \vdots \\ 0 & 0 & [0,..,0,K_{xrm}] \\ 0 & 0 & I \\ 0 & 0 & -I \end{bmatrix}}_{G_2} \underbrace{\begin{bmatrix} x_k \\ \underset{\rightarrow k}{c} \\ \underset{\rightarrow k+1}{r} \end{bmatrix}}_{Z_k} \leq \underbrace{\begin{bmatrix} \lambda\bar{u} \\ \vdots \\ \lambda\bar{u} \\ \lambda\underline{u} \\ \vdots \\ \lambda\underline{u} \\ \lambda\bar{x} \\ \vdots \\ \lambda\bar{x} \\ \bar{r} \\ \underline{r} \end{bmatrix}}_{f_2}. \tag{6.41}$$

These elements need to be tested only at the first sample ($k = 0$), so all constraints at each sample instant can be summarised using the following inequality:

$$\begin{bmatrix} G_1 \\ --- \\ G_2 \end{bmatrix} Z_k \leq \begin{bmatrix} f_1 \\ --- \\ f_2 \end{bmatrix}, \quad \forall k \tag{6.42}$$

where $G_1,\ G_2,\ f_1$ and f_2 are defined in (6.34) and (6.41)

Theorem 6.1 *One can deploy the algorithm of Section 4.5.2 with sample constraints (6.42) and autonomous model (6.25) and the algorithm will converge, provided that condition (6.3) is satisfied.*

Proof: It is known from condition (6.3), combined with the convergence to fixed values within n_c, n_a steps of states $\tilde{c}_k$, $\underset{\rightarrow k+1}{r}$, that the predictions of (6.25) must converge to a fixed steady-state. The algorithm of Section 4.5.2 shows therefore that asymptotically adding predictions for higher horizons results in redundant constraints beyond a certain horizon and therefore the algorithm will terminate. □

Now, it is straightforward to determine the MCAS in terms of a matrix F_r and a vector of limits d_r, defining the robust MCAS (RMCAS) in the form:

$$F_r Z_k \leq d_r \tag{6.43}$$

or in expanded form as:

$$S_{RMCAS} = \left\{ x : \exists \underset{\rightarrow k}{\tilde{c}} \quad s.t \quad M_r x_k + N_r \underset{\rightarrow k}{\tilde{c}} + V_r \underset{\rightarrow k+1}{r} \leq d_r \right\} \tag{6.44}$$

where M_r, N_r, V_r are suitable matrices and d_r is a vector of the limits.

6.4.4 Robust OMPC algorithm with tracking for reachable targets

In the previous subsection, we showed how to construct a robust MCAS of (6.44), for the tracking scenario. Now, we will use this MCAS in this subsection to summarise the proposed robust tracking MPC algorithm with advance knowledge for reachable targets as follows.

Algorithm 6.2 *Define the performance index as in (6.21). Define the robust MCAS as in (6.44). Perform the quadratic programming optimisation:*

$$\min_{\underset{\rightarrow k}{\tilde{c}}} \quad J \quad s.t. \quad M_r x_k + N_r \underset{\rightarrow k}{\tilde{c}} + V_r \underset{\rightarrow k+1}{r} \leq d_r \tag{6.45}$$

Implement the first block element of $\underset{\rightarrow k}{\tilde{c}}$ in (6.19) to compute the control law.

Theorem 6.2 *Algorithm 6.2 gives guaranteed convergence and recursive feasibility.*

Proof: By definition, the satisfaction of the RMCAS of (6.44) ensures recursive feasibility. Consequently, one can use conventional approaches [126] to show that $\tilde{c}_k$ converges to a weighted minimum. The convergence of $\tilde{c}_k$ implies convergence of the state x_k due to condition (6.3) and dynamics (6.25). □

6.5 Robust tracking MPC for unreachable targets

In the previous section, we discussed the MPC tracking of uncertain systems for scenarios where the targets are reachable in the steady-state. In this section, we will show how this robust tracking MPC can be extended in order to be implemented in the scenarios related to unreachable targets. The differences lie in the structures of the sample constraints components G_1, G_2 and transition matrices Ψ_i. Therefore, it is convenient to derive the components which are appropriate for this scenario.

6.5.1 Input parametrisation and cost function for unreachable targets

Analogous to Lemma 5.1, in order to deal with an unreachable target, we need to include the term c_∞ within the input parametrisation of (6.18). Therefore, it is convenient to represent the input predictions of the uncertain model for unreachable targets as:

$$\begin{cases} u_k - u_{ss|k} = -K(x_k - x_{ss|k}) + c_k, & k < n_c \\ u_k - u_{ss|k} = -K(x_k - x_{ss|k}) + c_\infty, & k \geq n_c \end{cases} \tag{6.46}$$

where c_∞ is assumed to be a slack variable, or an additional degree of freedom, and is well defined in Section 5.2.

It has also been shown that a suitable cost function, which trades off errors in the asymptotic target and the transient tracking errors can be defined as:

$$J_p = W_1(c_\infty^T S c_\infty) + \underset{\rightarrow k}{\tilde{c}}^T S_c \underset{\rightarrow k}{\tilde{c}} \tag{6.47}$$

where W_1 is a scalar weighting to be selected. Here, the term $(c_\infty^T S c_\infty)$ penalises asymptotic offset while the term $(\tilde{c}_{\rightarrow k}^T S \tilde{c}_{\rightarrow k})$ penalises transient performance, including information on $r_{\rightarrow k+1}$. The scalar weighting W_1 allows the users to determine how much emphasis they wish to place on each term.

As discussed in Section 6.4.1, it is possible to use the nominal cost function (6.47) for the uncertain case.

6.5.2 The closed-loop dynamics for unreachable targets

As discussed in Section 6.4.2, the uncertain model can be described in autonomous model formulation that captures the control perturbation $\tilde{c}$, c_∞ and the future target information $r_{\rightarrow k+1}$. This can be achieved by modifying $c_{\rightarrow k}$ in the dual-mode prediction of (6.22) as:

$$c_{\rightarrow k} = \tilde{c}_{\rightarrow k} + P_r r_{\rightarrow k+1} + c_\infty \tag{6.48}$$

Thus, the closed-loop dynamics can be described as:

$$x_{k+1+i} = \Phi(k)x_k + [B_i, \cdots 0]\tilde{c}_{\rightarrow k} + [[B_i, \cdots, 0]Pr + [(\Phi(k) - I)K_{xr}, \cdots, 0]] r_{\rightarrow k+1+i} \tag{6.49}$$

Similarly, the control input can be given by:

$$u_{k+i} = -Kx_{k+i} + [[I, \cdots 0]\tilde{c}_{\rightarrow k} + [I, \cdots, 0]Pr + [KK_{xr} + K_{ur}, \cdots, 0]] r_{\rightarrow k+1+i} \tag{6.50}$$

Analogous to Lemma 6.1, the uncertain dual-mode system predictions of (6.49) can be captured in a single mode autonomous model of the following form:

$$Z_{k+1} = \Psi(k)Z_k; \quad Z_k = [x_k^T, \tilde{c}_{\rightarrow k}^T, c_\infty, r_{\rightarrow k+1}^T]^T; \tag{6.51}$$

$$\Psi(k) = \begin{bmatrix} \Phi(k) & [B_i, 0, ..., 0] & 0 & [B(k), 0, ..., 0]Pr + [(\Phi(k) - I)K_{xr}, 0, .., 0] \\ 0 & D_C & E_C & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & D_R \end{bmatrix} \tag{6.52}$$

where $D_C,\ E_C$ are the shift matrices, defined in (5.12); $\Psi(k) \in Co\{\Psi_1, ...\Psi_m\}$.

Analogous to Lemma 6.1, the quadratic invariance of the closed-loop dynamic $\Phi(k)$ is sufficient to ensure the quadratic invariance of the augmented dynamic $\Psi(k)$.

Similarly, the input predictions of (6.50) can be described in terms of the autonomous model as:

$$u_k = -KZ_k \tag{6.53}$$

Remark 6.5 *Model predictions from (6.51) must converge to the specified artificial target of $\hat{r}_k$. The dynamics are known to be convergent. The asymptotic control law is defined as $u_k - u_{ss} = -K(x_k - x_{ss}) + c_\infty$ and, by definition, c_∞ is the value that ensures that the associated steady-state output is $\hat{r}_k$ (if $c_\infty = 0$, then the system converges to r_{k+n_a}).*

6.5.3 Derivation robust MCAS for unreachable targets

The derivation of robust MCAS was shown in Section 6.4.3, assuming that the target is reachable at steady-state. In this subsection, we will present a derivation of the robust MCAS for an unreachable target.

At this point, we consider the prediction model (6.51) and the transition matrix (6.52). It was shown in the previous section that one can define a robustly invariant set to guarantee robust stability for tracking scenarios, assuming that the target is reachable. This section extends this set for the case where the target is unreachable by deploying a similar concept but with the autonomous model of (6.52), which includes as states the degrees of freedom c_k, c_∞ and also the future target values $\underset{\rightarrow k+1}{r}$. A major difference is the convergence to a non-zero steady-state.

Following the same procedures as outlined in Section 6.4.3, the system constraints at each sample instant can be summarised with the following inequalities:

Transient constraints

The input and state constraints can be described in terms of the augmented states as:

$$\begin{bmatrix} -K & [I,0..,0] & 0 & [1,0,...,0]Pr+[KK_{xr}+K_{ur},0,..,0] \\ K & -[I,0..,0] & 0 & -[1,0,...,0]Pr+[KK_{xr}+K_{ur},0,..,0] \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{u} \\ \underline{u} \end{bmatrix}$$

and:

$$\begin{bmatrix} K_{xmax} & 0 & 0 & 0 \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{x} \end{bmatrix} \tag{6.54}$$

The expected steady-state constraints can also be described as:

$$\begin{bmatrix} 0 & 0 & 0 & [K_{ur},0,..0,] \\ 0 & 0 & 0 & [-K_{ur},0,..,0] \\ 0 & 0 & 0 & [K_{xr},0,..0,] \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{u} \\ \underline{u} \\ \bar{x} \end{bmatrix} \tag{6.55}$$

Combining the inequalities of (6.54) and (6.55), we can then define G_1 as:

$$\underbrace{\begin{bmatrix} -K & [I,0..,0] & 0 & [1,0,...,0]Pr+[KK_{xr}+K_{ur},0,..,0] \\ K & -[I,0..,0] & 0 & -[1,0,...,0]Pr+[KK_{xr}+K_{ur},0,..,0] \\ 0 & 0 & 0 & [K_{ur},0,..0,] \\ 0 & 0 & 0 & [-K_{ur},0,..,0] \\ 0 & 0 & 0 & [K_{xr},0,..0,] \\ K_{xmax} & 0 & 0 & 0 \end{bmatrix}}_{G_1} Z_k \leq \underbrace{\begin{bmatrix} \bar{u} \\ \underline{u} \\ \bar{u} \\ \underline{u} \\ \bar{x} \\ \bar{x} \end{bmatrix}}_{f_1} \tag{6.56}$$

or, in compact form as:

$$G_1 Z_k \leq f_1 \tag{6.57}$$

Steady state constraints

Steady state and input constraints can be described as:

$$\begin{bmatrix} 0 & 0 & K_{uri} & [0,..0,K_{uri}] \\ 0 & 0 & -K_{uri} & [0,..,0,-K_{uri}] \\ 0 & 0 & K_{xri} & [0,..,0,K_{xri}] \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{u} \\ \underline{u} \\ \bar{x} \end{bmatrix}, \quad i=1,\cdots\cdots,m \tag{6.58}$$

The asymptotic target limits can be given as:

$$\begin{bmatrix} 0 & 0 & 0 & I \\ 0 & 0 & 0 & -I \end{bmatrix} Z_k \leq \begin{bmatrix} \bar{r} \\ \underline{r} \end{bmatrix} \tag{6.59}$$

Combining the sample constraints of (6.58) with the asymptotic target of (6.59), G_2 can be defined as:

$$\underbrace{\begin{bmatrix} 0 & 0 & K_{ur1} & [0,..0,K_{ur1}] \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & K_{urm} & [0,..0,K_{urm}] \\ 0 & 0 & -K_{ur1} & [0,..,0,-K_{ur1}] \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -K_{urm} & [0,..,0,-K_{urm}] \\ 0 & 0 & K_{xr1} & [0,..,0,K_{xr1}] \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & K_{xrm} & [0,..0,K_{xrm}] \\ 0 & 0 & 0 & I \\ 0 & 0 & 0 & -I \end{bmatrix}}_{G_2} Z_k \leq \underbrace{\begin{bmatrix} \bar{u} \\ \vdots \\ \bar{u} \\ \underline{u} \\ \vdots \\ \underline{u} \\ \bar{x} \\ \vdots \\ \bar{x} \\ \bar{r} \\ \underline{r} \end{bmatrix}}_{f_2} \tag{6.60}$$

or, in compact form:

$$G_2 Z_k \leq f_2 \tag{6.61}$$

It is convenient to combine (6.57) and (6.61) together, to be defined in a single form as:

$$\begin{bmatrix} G_1 \\ --- \\ G_2 \end{bmatrix} Z_k \leq \begin{bmatrix} f_1 \\ --- \\ f_2 \end{bmatrix}, \quad \forall k \tag{6.62}$$

Now, it is straightforward to define the robust MCAS for unreachable targets by using Algorithm 2.1 with equations of (6.51) and (6.62), as follows.

$$F_r Z_k \leq d_r \tag{6.63}$$

or in expanded form as:

$$\tilde{S}_{RMCAS} = \left\{ x : \exists \tilde{c}_k, c_\infty \quad s.t \quad M_r x_k + N_r \underset{\rightarrow k}{\tilde{c}} + Q_r c_\infty + V_r \underset{\rightarrow k+1}{r} \leq d_r \right\} \tag{6.64}$$

where $M_r,\ N_r,\ Q_r$ and V_r are suitable matrices and d_r is a vector of the limits.

6.5.4 Robust OMPC algorithm with tracking for unreachable targets

In the previous subsection, we defined the MCAS (6.64) for uncertain systems for unreachable targets. Now, we will use this MCAS in this subsection to summarise the proposed robust tracking MPC algorithm with advance knowledge for unreachable targets as follows.

Algorithm 6.3 *Define the performance index as in (6.47). Define the robust MCAS as in (6.64). Perform the quadratic programming optimisation:*

$$\min_{c_\infty, \tilde{c}_k} \quad J \quad s.t. \quad M_r x_k + N_r \underset{\rightarrow k}{\tilde{c}} + Q_r c_\infty + V_r \underset{\rightarrow k+1}{r} \leq d_r \tag{6.65}$$

Implement the first block element of $\tilde{c}_k$ in (6.48) to compute the control law, as defined in (6.46).

Theorem 6.3 *Algorithm 6.3 gives guaranteed convergence and recursive feasibility, including cases of unreachable set points.*

Proof: By definition, the satisfaction of the RMCAS of (6.64) ensures recursive feasibility. Consequently, one can use conventional approaches [126] to show that $\tilde{c}_k$ converges to a weighted minimum. The convergence of $\tilde{c}_k$ implies convergence of the state x_k due to condition (6.3) and dynamics (6.52). □

6.6 Summary: Robust to parameter uncertainty

1. The model predictions take the form:

$$Z_{k+1} = \Psi(k) Z_k \tag{6.66}$$

2. The input and state/output constraints take the form:

$$G_1 Z_k \le f_1; \quad G_2 Z_k \le f_2 \tag{6.67}$$

3. For a reachable target, the robust MCAS (denoted S_{RMCAS}) can be defined as:

$$S_{RMCAS} = \left\{ x : \exists \underset{\rightarrow k}{\tilde{c}} \;\; s.t \;\; M_r x_k + N_r \underset{\rightarrow k}{\tilde{c}} + V_r \underset{\rightarrow k+1}{r} \le d_r \right\} \tag{6.68}$$

4. For a unreachable target, the robust MCAS (denoted $\tilde{S}_{RMCAS}$) can be defined as:

$$\tilde{S}_{RMCAS} = \left\{ x : \exists \tilde{c}_k, c_\infty \;\; s.t \;\; M_r x_k + N_r \underset{\rightarrow k}{\tilde{c}} + Q_r c_\infty + V_r \underset{\rightarrow k+1}{r} \le d_r \right\} \tag{6.69}$$

6.7 Numerical illustrative examples

This section demonstrates that Algorithms 6.2 and 6.3 are both robust to parameter uncertainty and handle advance information about target changes effectively. Conversely, an algorithm which does not embed the parameter uncertainty gives less effective performance and indeed could lose feasibility. We will present two examples with different dynamics to show the efficacy and benefits of the proposed algorithms compared to those that already exist in the literature. In particular, these examples show how effectively the proposed algorithms handle both reachable and unreachable targets through the systematic use of advance knowledge.

6.7.1 Example: 1

For ease of comparison, we will use in this example the uncertain system model that was presented in [83] since it has originally non-linear properties, as captured by the LPV model.

The matrices of the model are given by:

$$A = Co\left\{\begin{bmatrix} 0.807 & -0.0037 \\ 21.8638 & 1.4936 \end{bmatrix}, \begin{bmatrix} 0.8326 & -0.0029 \\ 16.902 & 1.3291 \end{bmatrix}\right\}; B = Co\left\{\begin{bmatrix} -0.0003 \\ 0.2153 \end{bmatrix}, \begin{bmatrix} -0.0002 \\ 0.2022 \end{bmatrix}\right\} \tag{6.70}$$

where

$$A_1 = \begin{bmatrix} 0.807 & -0.0037 \\ 21.8638 & 1.4936 \end{bmatrix}; \quad A_2 = \begin{bmatrix} 0.8326 & -0.0029 \\ 16.902 & 1.3291 \end{bmatrix}; B_1 \begin{bmatrix} -0.0003 \\ 0.2153 \end{bmatrix}; \quad B_2 = \begin{bmatrix} -0.0002 \\ 0.2022 \end{bmatrix}$$

The system input/state constraints are:

$$\begin{matrix} -2.0 \leq u \leq 2.0; \\ -0.5 \leq u_{ss} \leq 0.5; \end{matrix} \quad \begin{bmatrix} -0.5 \\ -5 \end{bmatrix} \leq x \leq \begin{bmatrix} 0.5 \\ 5 \end{bmatrix} \tag{6.71}$$

A nominal model is assumed to be $A = 0.6A_1 + 0.4A_2$, and $B = 0.6B_1 + 0.4B_2$ is used to define the feedback controller $K = \begin{bmatrix} 98.4074 & 4.0843 \end{bmatrix}$ as the LQ-Optimal for $Q = C^T C$ and $R = 0.1I$.

Case 1: Robust OMPC for reachable targets with advance knowledge

This case presents Algorithm 6.3 for the LPV system (6.70) in which the advance knowledge of $n_a = 2$ is considered for the robust tracking MPC optimisation.

Figure 6.2 shows the output response with the use of advance knowledge, $n_a = 2$ and a target, $r = 1$, which is reachable at the steady-state but unreachable in the transients.

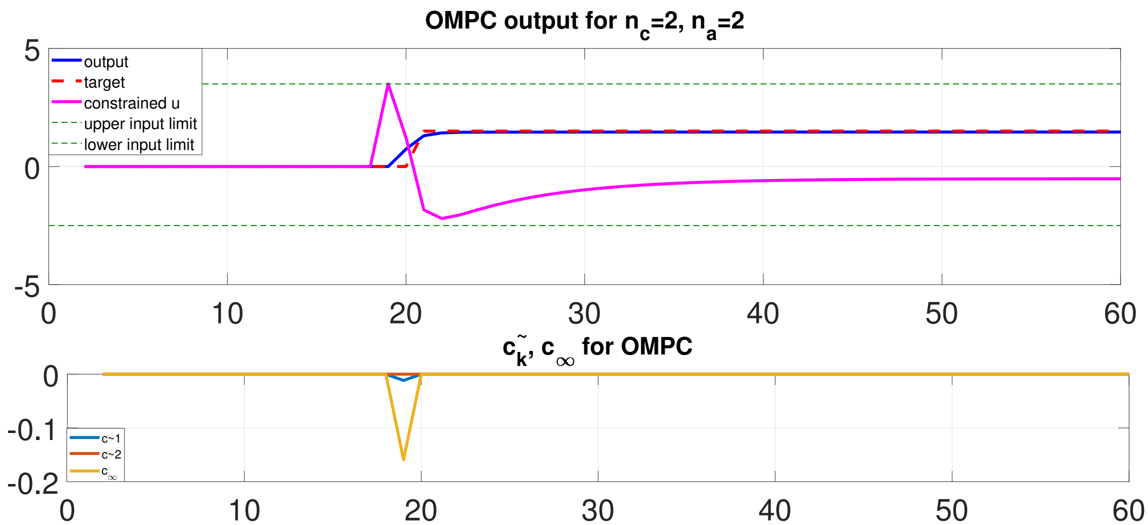

Figure 6.2: Closed-loop response of system (6.70) for reachable target with advance knowledge $n_a = 2$.

It is shown that, although the target is unreachable during transients, the robust MPC algorithm performs well in handling the advance target information effectively and avoiding constraint violations. The output smoothly converges to the correct steady state. It will be noted that the perturbation terms c_∞ and $\tilde{c_k}$ are non-zero in the transients only and become zero for a longer time. The term c_k is non-zero, as expected from equation (6.48).

Case 2: Robust OMPC tracking for unreachable targets with advance knowledge

This case presents the scenario in which the target is unreachable during transient and steady-states.

Figure 6.3 shows the output response of the LPV system (6.70) with the use of an advance knowledge, $n_a = 2$, and a target, $r = 1.3$, which is unreachable at steady-state.

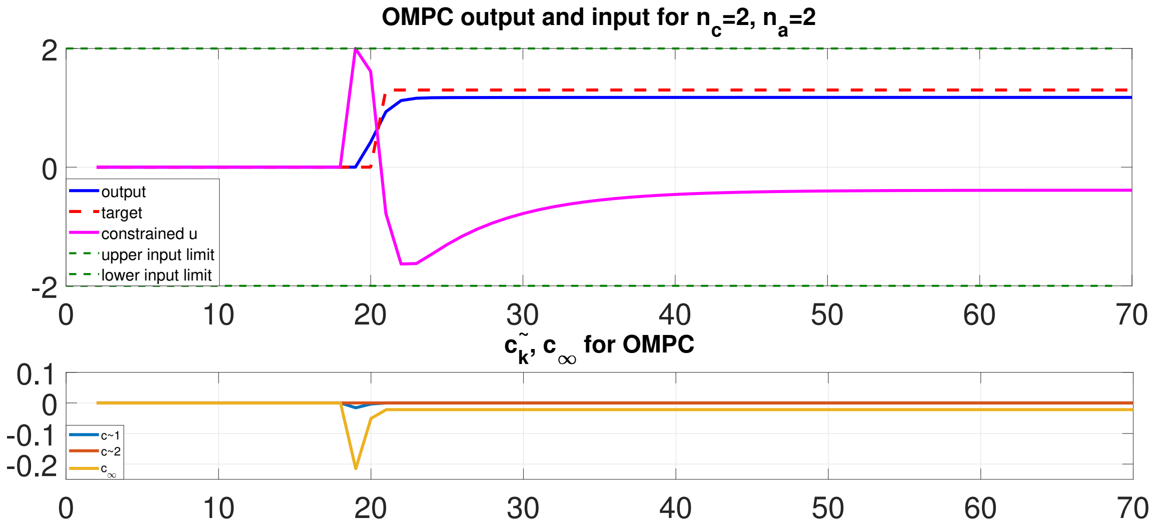

Figure 6.3: Closed-loop for the step response of system (6.70) for unreachable targets with advance knowledge $n_a = 2$.

It can be seen that, although the target is unreachable during both the transients and the steady states, the robust MPC algorithm performs well, handling both the advance target information effectively while the output converges as close as possible to the correct steady-state without any constraint violations. It will be noted that the perturbation term c_∞ is non-zero for the long term whereas $\tilde{c_k}$ is non-zero in the transients only since the input constraints are active in the transients. The term c_k is non zero, as expected from equation (6.48).

6.7.2 Example: 2

In order to ensure the efficacy of the proposed algorithm, we will consider a larger dimensional unstable LPV system, which is adopted from [88], to demonstrate also that the proposed algorithm can effectively handle the uncertainty even for large, unstable systems.

The matrices of the model are given by:

$$A = Co\left\{\begin{bmatrix} 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 0.1 & 0 & 0 & 0 \\ 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 0.1 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 1 & 0 & 1 & 0 \\ 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 1 & 0 & 1 & 0 \end{bmatrix}\right\} \tag{6.72}$$

where

$$A_1 = \begin{bmatrix} 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 0.1 & 0 & 0 & 0 \\ 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 0.1 & 0 & 0 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 1 & 0 & 1 & 0 \\ 1.3333 & -0.6667 & 1.3333 & -0.6667 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

and

$$B_1 = B_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The system input/state constraints are:

$$-1 \le u_k \le 1; \quad \begin{bmatrix} -1.15 \\ -1.15 \\ -1.15 \\ -1.15 \end{bmatrix} \le x_k \le \begin{bmatrix} 1.15 \\ 1.15 \\ 1.15 \\ 1.15 \end{bmatrix} \tag{6.73}$$

A nominal model is assumed to be $A = 0.6A_1 + 0.4A_2$, and $B = B_1 = B_2$ is used to define the feedback controller $K = \begin{bmatrix} 1.7698 & -1.1060 & 1.8100 & -1.1060 \end{bmatrix}$ as the LQ-Optimal for $Q = diag([1, 1, 1, 1])$ and $R = 0.1I$.

Case 1: Robust OMPC tracking for reachable targets with advance knowledge

In this case, we assume a target of $r = 0.1$ and choose advance knowledge of $n_a = 2$ with $n_c = 2$.

Figure 6.4 shows the OMPC output response of LPV system (6.72) with advance knowledge for a reachable target.

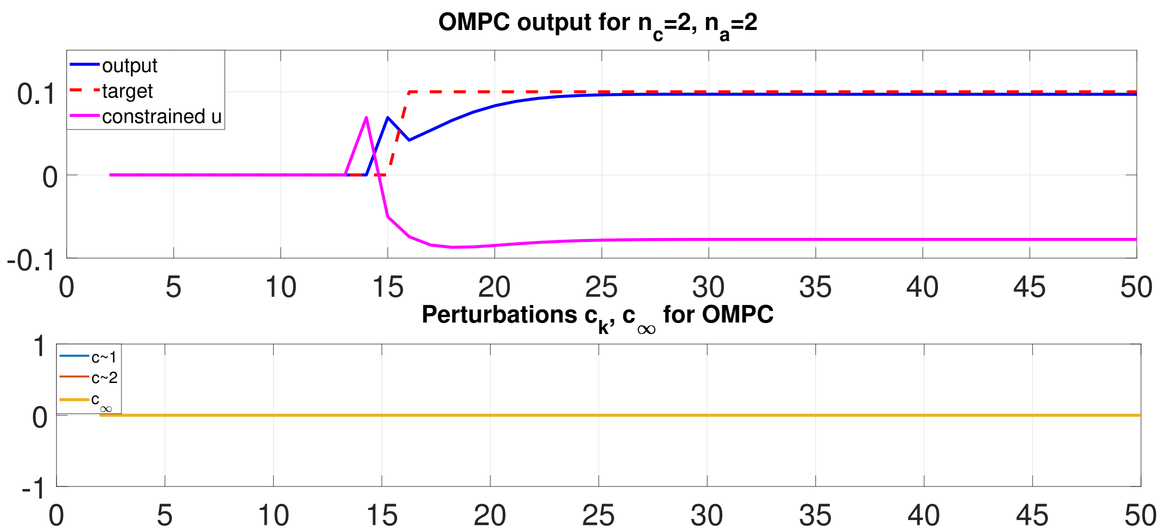

Figure 6.4: Closed-loop for the step response of system (6.72) for a reachable target with $n_a = 2$.

It is shown in Figure 6.4 that the target ($r = 0.1$) is reachable during both the transients and steady-states and that the robust MPC algorithm performs well and converges to the correct steady-state without any constraint saturation. It will be noted that the perturbation terms c_∞ and $\tilde{c_k}$ are zero for a long time since there no deviations about either the unconstrained optimal or the true target. However, the perturbation c_k of (6.48) is non-zero, as expected due to the inclusion of advance information about the target changes.

Case 2: Robust OMPC tracking for unreachable targets with advance knowledge

In this case, we assume that the target is unreachable at steady-state; that is, ($r = 0.12$) with advance knowledge, $n_a = 2$, and a control horizon, $n_c = 2$.

Figure 6.5 shows the OMPC output response of the LPV system (6.72) with advance knowledge for unreachable target.

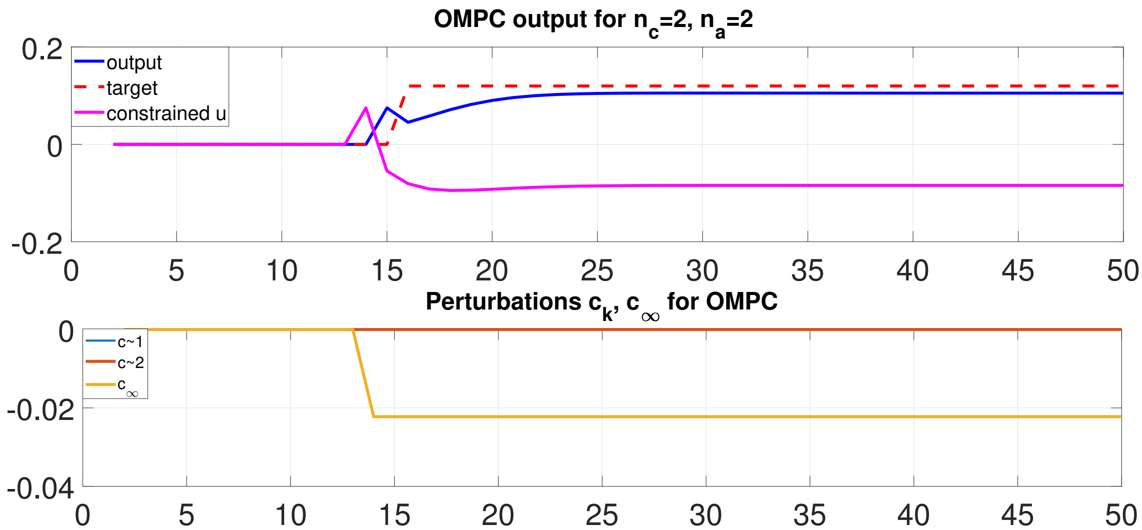

Figure 6.5: Closed-loop for the step response of system (6.72) for unreachable target with $n_a = 2$.

It is shown in Figure 6.5 that, although the target ($r = 0.12$) is unreachable, nevertheless the robust MPC algorithm performs well and converges to the correct steady-state without any constraint violations. It will be noted that ($c_\infty \neq 0$) for a long time, whereas $\tilde{c_k} = 0$ for a long time, since the input constraints are inactive, but there exist deviation from the true target. However, the perturbation c_k of (6.48) is non-zero, as expected, due to the inclusion of advance information about the target changes.

The benefits of using advance knowledge are further evidenced in Table 6.1, which compares the performance indices of the corresponding systems for reachable targets.

Table 6.1 shows the performance indices of systems (6.70) and (6.72) with and without advance knowledge for both reachable targets.

	J with $n_a = 1$	J with $n_a = 3$
System (6.70) for reachable targets	1.262	0.256
System (6.72) for reachable targets	1.55	0.527

Table 6.1: Performance indices for step changes in the target for system (8.14)

It is clear that the performance indices for the systems with advance knowledge are lower than those for the systems without advance knowledge. This provides evidence of the advantages of using advance knowledge in robust OMPC algorithms.

6.8 Conclusion

This chapter made a new contribution to this field, which is the proposal of an efficient, robust MPC algorithm which systematically incorporates advance information about the target. Most of the robust approaches in the literature either focus on the regulation case or ignore feed-forward information. Robust feasibility and convergence have been achieved. Moreover, it has been demonstrated that algorithms for incorporating unreachable set points can also be modified in a straightforward manner for the robust case.

It has been shown that the robust MCAS can be efficiently implemented in the LPV systems for both the regulation and tracking MPC problems. The differences occur only in the autonomous model structure and hence in the sample constraints.

The results of robust MPC tracking for both reachable and unreachable targets have been discussed in this chapter. It is noted that those results are similar in terms of the procedures for deriving the MCAS for uncertain systems and the description of model dynamics and input parametrisation.

On the other hand, these results differ in terms of the definition of the augmented states, as the latter contains two degrees of freedom $\tilde{c}_k, c_\infty$ while the former includes only one degree of freedom $\tilde{c}_k$. Numerical examples demonstrate the benefits of using future information as well as the ability to deal with either temporary or permanently unreachable targets. Further illustration is presented in Chapter 8.

Chapter 7

IMPROVING PARAMETRIC APPROACHES WITHIN MPC TRACKING

This chapter discusses two issues. One is related to what extent the future values of the target can be treated as states in a parametric optimization. The other important, and linked, issue is to optimise the artificial target and investigate the impact of the input parametrisation on predictive control using the parametric approach.

7.1 Introduction

One of the most significant advances in predictive control of the past 20 years has been the recognition that one can define the solution of quadratic programming (QP), in full, using off-line computations [112, 15]. As long as this off-line (or so-called parametric solution) is not too complex, then coding and implementing this on-line may be far simpler than implementing an on-line QP optimiser. The parametric solution offers the potential for reliability, transparency (important for validation and certification) and most importantly, very fast sample rates for some systems.

Nevertheless, parametric solutions also have their disadvantages and the literature is full of possible solutions to counter these [72]. For example: (i) in some cases, the parametric solution can be difficult to compute reliably due to poor conditioning; (ii) where the parametric solution requires large numbers of regions, it may no longer be computationally efficient.

This chapter makes some contributions to one aspect of computational complexity. To define what this contribution will be, it is first useful to define a generic QP optimisation problem and its parametric solution.

$$\min_z z^T S z + z^T P w \quad \text{s.t.} \quad Nz + Mw \leq d \tag{7.1}$$

where w is a system state, z are the degrees of freedom (d.o.f.) and parameters S, P, N, M, d define the cost function and linear constraints. A parametric solution (often denoted as mp-QP) partitions the space into a number of non-overlapping regions for the system state such that the optimal solution for (7.1) is equivalent to:

$$N_i w \leq d_i \quad \Rightarrow \quad z = K_i w + p_i, \quad i = 1, 2, \cdots, n \tag{7.2}$$

for suitable N_i, d_i, K_i, p_i.

It is recognised that there is a strong link between the dimension of the state w and the required number of regions n to capture the entire solution. Hence, in general, parametric solutions are favoured for systems with a low state dimension but less likely to be useful for high state dimensions. Moreover, the higher the state dimension, the more likely one is to encounter conditioning problems in the mp-QP solver.

It is also recognized that the parametric solutions are very complex, requiring excessive off-line computation, excessive on-line storage and set membership tests [72]. Thus, there is interest, in finding alternatives which provide simple solutions, perhaps at the cost of some sub-optimality [12].

According to the above discussion, this chapter, therefore, makes two contributions. One contribution is to ask the question: can we reduce the state dimension for some specific predictive control problems? In particular, the focus here is on the handling of feed-forward information such as future target information which, in principle, can be embedded systematically in predictive control algorithms. The other contribution is to investigate the potential of different input parametrisations to reduce the complexity of parametric solutions alongside a simple aim of achieving sufficiently large feasible regions [141].

Section 7.2 outlines the OMPC algorithm to be implemented with parametric approaches. Section 7.3 will demonstrate how the state dimension can be reduced using some elementary algebra. Section 7.4 demonstrates the influence of future target information on the parametric complexity. Section 7.5 investigates the impact of steady state offset on the parametric approaches. Section 7.6 discusses the impact of the steady state offset on the feasible regions. Section 7.7 presents some numerical examples on feasibility with the parametric solutions. Section 7.8 demonstrates the simplification of the parametric solutions complexity. Finally, the chapter finishes by presenting the conclusions.

7.2 Basics of the dual-mode (OMPC) approach

This section summarises briefly the optimisation implicit in optimal predictive control (OMPC) [146], [153]. A specific and important point is to consider how set point information is incorporated. This detail is presented in [36, 37, 166], and discussed in detail in Chapter 4, Section 4.3.

For simplicity, this chapter assumes a state space model of the following form:

$$x_{k+1} = Ax_k + B\Delta u_k; \quad y_k = Cx_k \tag{7.3}$$

with x_k, u_k, y_k the states, input and output respectively, with dimensions n_x, m, m and $\Delta u_k = u_k - u_{k-1}$. The system is subject to constraints, typically (others are possible):

$$\underline{u} \le u_k \le \overline{u}; \quad \underline{\Delta u} \le \Delta u_k \le \overline{\Delta u}; \quad \underline{x} \le K_x x_k \le \overline{x} \tag{7.4}$$

Furthermore, define the future target $\underset{\rightarrow k+1}{r}$ as follows.

$$\underset{\rightarrow k+1}{r} = [r_{k+1}^T, r_{k+2}^T, \cdots, r_{k+n_y}^T]^T \tag{7.5}$$

The system steady-state states and inputs are estimated by solving $x_{k+1} = x_k,\ y_k = r_{k+1}$ and hence:

$$\begin{bmatrix} A-I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} 0 \\ r_{k+1} - d_k \end{bmatrix} \tag{7.6}$$

$$\rightarrow \quad \begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} K_{xr} \\ K_{ur} \end{bmatrix} (r_{k+1} - d_k)$$

where K_{xr}, K_{ur} are well known in [101].

7.2.1 OMPC with a parametric approach

It is now well known [95] that the dual-mode (OMPC) approaches have efficient stability properties in general. Consequently, it is worth considering how a parametric solution can be determined for a dual-mode approach. Thus, it is convenient to restate the standard OMPC algorithm which, is utilised in [146] and was discussed in Chapter 4, before formulating a parametric problem for tracking scenarios.

It was shown in Section 4.3 that target future information can be used wisely in predictive control, as this detail is neglected in the mainstream literature. In this chapter, we will show how this future information can be augmented to the system state and how it influences the dimension of the parametric space. To remind the reader, we will outline the core components of the OMPC algorithm with the inclusion of future target information as follows.

OMPC uses an infinite horizon performance index of the following form:

$$J = \sum_{i=0}^{\infty}(x_{k+1+i} - x_{ss|k+1+i})^T Q (x_{k+1+i} - x_{ss|k+1+i}) + (u_{k+i} - u_{ss|k+i})^T R (u_{k+i} - u_{ss|k+i}) \tag{7.7}$$

along with an input parametrisation of the form:

$$\begin{aligned} u_{k+i} - u_{ss|k+i} &= -K(x_{k+i} - x_{ss|k+1+i}) + c_{k+i}, \quad && i = 0, 1, \cdots, n_c - 1 \\ u_{k+i} - u_{ss|k+i} &= -K(x_{k+i} - x_{ss|k+1+i}), && i \geq n_c \end{aligned} \tag{7.8}$$

so the variables c_{k+i} are the degrees of freedom and (x_{ss}, u_{ss}) are the expected steady-states for tracking a fixed target, r_{k+1}.

Extending the dual-mode strategies to take account of more future values of the target such as available in $\underset{\rightarrow k+1}{r}$ is discussed in Chapter 4 Theorem 4.1 and [36]. On this point, it is shown that minimising the performance index (7.7) is equivalent to minimising the following form:

$$J = \underset{\rightarrow k}{\tilde{c}}^T S_c \underset{\rightarrow k}{\tilde{c}} \tag{7.9}$$

where the term $\underset{\rightarrow k}{\tilde{c}}$ is the deviation from the unconstrained optimum in the tracking case, such that the control input is $u_k = -Kx_k + \underset{\rightarrow k}{\tilde{c}} + P_r \underset{\rightarrow k+1}{r}$.

It was also shown in Subsection 4.5.2 that constraint inequalities representing constraint satisfaction of the predictions can be reduced to:

$$N \underset{\rightarrow k}{\tilde{c}} + Mx_k + Q \underset{\rightarrow k+1}{r} \leq d \tag{7.10}$$

for suitable N, M, Q and d is a vector of the limits.

It is straightforward to define an optimisation problem by combining (7.9,7.10), to take the following form:

$$\min_{\underset{\rightarrow k}{\tilde{c}}} \underset{\rightarrow k}{\tilde{c}}^T S \underset{\rightarrow k}{\tilde{c}} \quad \text{s.t.} \quad N \underset{\rightarrow k}{\tilde{c}} + Mx_k + Q \underset{\rightarrow k+1}{r} \leq d. \tag{7.11}$$

Now, it is possible to define the optimisation problem (7.11) as a parametric problem, as shown in the following lemma:

Lemma 7.1 *Optimisation problem (7.11) can be recast in the same form as (7.1).*

Proof: By combining parameters which may vary with time (that is $\underset{\rightarrow k+1}{r}, x_k$), the optimisation problem (7.11) can be reformulated as follows.

$$\min_{\underset{\rightarrow k}{\tilde{c}}} \underset{\rightarrow k}{\tilde{c}}^T S_c \underset{\rightarrow k}{\tilde{c}} \quad \text{s.t.} \quad N \underset{\rightarrow k}{\tilde{c}} + [M, Q] \underbrace{\begin{bmatrix} x_k \\ \underset{\rightarrow k+1}{r} \end{bmatrix}}_{w_k} \leq d \tag{7.12}$$

for suitable N, M, Q.

Corollary 7.1 *Including advance knowledge $\underset{\rightarrow k+1}{r}$ in the OMPC algorithm augments the implied parametric state dimension by the dimension of $\underset{\rightarrow k+1}{r}$ to give a dimension of $(n_x + n_y \times m)$. This is obvious from the replacement of x_k by w_k in (7.12).*

Now, the obvious dilemma is that incorporating advance knowledge of the target into a predictive control problem increases the dimension of the parametric state w_k. Even simply including a fixed target and integral action [150] increases the required parameter dimension to (n_x+m), which is already undesirable, so including more advance information could make the dimension of the parametric optimisation impractical in general.

Hence, one objective of this chapter is to suggest ways of modifying optimisation (7.12) so that one retains some of the benefits of including advance information while at the same time keeping the dimension of the implied parameter space small.

7.2.2 Summary and proposals

It has been shown that a simplistic inclusion of future target information $\underset{\rightarrow k+1}{r}$ into a predictive control algorithm leads to an increase in the dimension of the parametric space for the associated multi-parametric quadratic programming. In general, for anything other than the most trivial case [150], where it is assumed that $r_{k+i} = r_{k+1}, \forall i$ so that the effective dimension of $\underset{\rightarrow k+1}{r}$ is reduced to just r_{k+1} , then this increase in dimension is likely to be unmanageable and thus a parametric approach is unlikely to be feasible. In consequence, this chapter considers into which scenarios this information can be incorporated without leading to unnecessarily large dimensional increases.

7.3 Reducing the dimension of the parameter space with OMPC algorithms

This section will show how small changes to the formulation of OMPC optimisation can reduce the dimension of the implied parameter space. Some suggestions lead to a small degree of sub-optimality but, in fact, within parametric predictive control, the use of sub-optimality is often a key tool for reducing the complexity [20, 12] and thus this may be considered an acceptable compromise in order to gain some of the benefits of using the feed-forward information rather than ignoring it.

In this manner, we will propose three alternatives which are available to the OMPC approach in the next subsections.

7.3.1 Reducing the amount of advance knowledge

Throughout this thesis, the use of advance information ([37] and Chapter 4) within predictive control has considered questions about how much advance information is useful; that is, really makes a noticeable difference to closed-loop performance. It was established in [37] that ignoring the far future (beyond n_a samples) values of the target usually led to a minimal deterioration in performance , provided that $n_u < n_a < (n_u + n_r)/2$, with n_r being the notional rise time. A larger n_a were usually unhelpful as the degrees of freedom were not contemporaneous enough and therefore inappropriate control moves for the relevant target changes. Therefore, as in the context of parametric approaches, some sub-optimality is acceptable in the pursuit of simplicity, so this section examines what can be achieved by summarising future target information $\underset{\to k+1}{r}$ into fewer values.

The most obvious and easiest way to reduce the dimension of the parameter space in vector w is the rather obvious one of reducing the dimension of $\underset{\to k+1}{r}$.

This section exploits some of the results from Chapter 4. It has been shown that it is useful to the use the following assumption $r_{k+n_a+i} = r_{k+n_a}, \forall i > 0$ (here the SISO case is given to simplify algebra). Then:

$$\underset{\to k+1}{r} = \begin{bmatrix} r_{k+1} \\ : \\ r_{k+n_a} \\ : \\ r_{k+n_a} \end{bmatrix} = \begin{bmatrix} r_{k+1} \\ r_{k+2} \\ : \\ : \\ L r_{k+n_a} \end{bmatrix}; L_{n_y - n_a} = \begin{bmatrix} 1 \\ 1 \\ : \\ 1 \end{bmatrix} \tag{7.13}$$

where L is a dimensional vector of ones. This assumption reduces the dimension of $\underset{\to k+1}{r}$ from n_y components to n_a components. Moreover, it has been shown in ([166, 36] and Chapter 4) that, in many cases, using relatively small values of n_a give almost equivalent, and sometimes better, closed-loop performance compared to using values close to n_y. Thus, using the approximation implicit in a choice of small n_a is reasonable for the OMPC approach.

Remark 7.1 *Even though one can reduce the overall parameter dimension of* $w = [x^T, \underset{\to k+1}{r}^T]^T$ *to* $n_x + n_a \times m$ *with* $n_a < n_y$*, one might still argue that anything much beyond* $n_a = 2$ *is likely to increase the parameter space beyond normally accepted limits for parametric solutions. While* $n_a = 2$ *usually gives better closed-loop performance than* $n_a = 1$*, nevertheless it may still be significantly worse performance than achievable with an even larger* n_a *and* *thus such a solution may not be sufficient in general.*

7.3.2 Reducing the amount of advance knowledge further

The existing literature has largely focused on the structure of (7.13), arguing that $n_a \ll n_y$ often leads to improved closed-loop behaviour [166, 36]. However, there is an alternative that has not been explored in depth and that is the subject of the current investigation. This is the extent to which transient values, such as r_{k+1}, r_{k+2}, are really useful as most systems cannot respond significantly within a few samples, so having a particular target during fast transients may not be meaningful. The proposal here, therefore, is to ignore specific information about the targets for the next few samples and instead assume that $r_{k+i} = r_{k+n_{a1}}, i \leq n_{a1}$. Thus we can use a structure such as the following:

$$\underset{\rightarrow k+1}{r} = \begin{bmatrix} r_{k+n_{a1}} \\ \vdots \\ r_{k+n_{a1}} \\ r_{k+n_{a1}+1} \\ \vdots \\ r_{k+n_a-1} \\ r_{k+n_a} \\ \vdots \\ r_{k+n_a} \end{bmatrix} = \begin{bmatrix} L_{n_{a1}} r_{k+n_{a1}} \\ r_{k+n_{a1}+1} \\ \vdots \\ r_{k+n_a-1} \\ L_{n_y-n_a} r_{k+n_a} \end{bmatrix} \tag{7.14}$$

It is clear that the dimension of the corresponding vector $\underset{\rightarrow k+1}{r}$ has now been reduced to having $(n_a - n_{a2} + 1)$ independent components, which is a significant reduction compared to n_y components.

7.3.3 Using insights from the reference governors and predictive function control

Reference governors [51] are primarily focused on highly efficient constraint handling whereby one ensures that the target of the feedback loop changes sufficiently slowly to avoid causing the internal signals to violate the constraints. To some extent, performance takes second place to computational efficiency and simplicity, so some sub-optimality is acceptable. In the context of this chapter, a key observation is the use of small amounts of feed-forward information about the target rather than the entire trajectory. Specifically, this chapter notes one possible simplification which is implicit in the Predictive Function Control [132]; that is, we assume the future target trajectory takes the following form (a smooth transition from the current output to the long term target):

$$r_{k+i} = (r_{k+n_a} - y_k)(1 - \lambda^i) + y_k; \tag{7.15}$$

Equation (7.15) can be simplified as:

$$r_{k+i} = (1 - \lambda^i) r_{k+n_a} + \lambda^i y_k \tag{7.16}$$

Now, future target information $\underset{\rightarrow k+1}{\hat{r}}$ for the first few samples n_{a1} can be expressed as:

$$\underbrace{\begin{bmatrix} r_{k+1} \\ r_{k+2} \\ \vdots \\ r_{k+n_{a1}-1} \\ r_{k+n_{a1}} \end{bmatrix}}_{\underset{\rightarrow k+1}{\hat{r}}} = \underbrace{\begin{bmatrix} (1-\lambda) \\ (1-\lambda^2) \\ \vdots \\ (1-\lambda^{n_{a1}-1}) \\ (1-\lambda^{n_{a1}}) \end{bmatrix}}_{W_a} r_{k+n_{a1}} + \underbrace{\begin{bmatrix} (\lambda) \\ (\lambda)^2 \\ \vdots \\ \lambda^{n_{a1}-1} \\ \lambda^{n_{a1}} \end{bmatrix}}_{W_b} y_k \tag{7.17}$$

Equation (7.17) can be described in a compact form as:

$$\underset{\rightarrow k+1}{r} = W_a r_{k+n_a} + W_b y_k \tag{7.18}$$

where the definitions of W_a, W_b are obvious and r_{k+n_a} is the best representation of the long term target value.

Clearly, this suggestion has close analogies to the previous two subsections, since the future target information is approximated in some fashion to reduce the dimension of the implied parametric space. The proposal here offers the advantage that the parametric space is the same dimension as would be needed for the routine inclusion of integral action [150], although of course the use of the feed-forward information is now far less precise than it could be due to the approximation implicit in (7.18).

The expression (7.18) is one example of a possible simple approximation. If we can allow a higher dimension, we can derive an alternative approximation as follows.

It is convenient to split the future target information into two parts. One part describes the first few samples of the target information, while the other describes the rest of the target information.

At this point, we can define the future target information as:

$$\underset{\rightarrow k+1}{r} = \underbrace{\begin{bmatrix} r_{k+1} \\ \vdots \\ r_{k+n_{a1}} \\ --- \\ r_{k+n_{a1}+1} \\ \vdots \\ r_{k+n_a} \end{bmatrix}}_{\underset{\rightarrow k+1}{r}} \tag{7.19}$$

Combining equations (7.18, 7.19) with some algebra, one can rewrite the future target information in the form:

$$\underbrace{\begin{bmatrix} r_{k+1} \\ r_{k+2} \\ \vdots \\ r_{k+n_{a1}} \\ r_{k+n_{a1}+1} \\ \vdots \\ r_{k+n_a} \end{bmatrix}}_{\underset{\rightarrow k+1}{r}} = \underbrace{\begin{bmatrix} 1-\lambda^1 & 0 & 0 & \cdots & 0 \\ 1-\lambda^2 & 0 & 0 & \cdots & : \\ : & : & : & : & : \\ 1-\lambda^{n_{a1}} & 0 & 0 & \cdots & : \\ 0 & I & 0 & \cdots & 0 \\ : & : & : & \cdots & 0 \\ 0 & 0 & 0 & \cdots & I \end{bmatrix}}_{W_c} \underbrace{\begin{bmatrix} r_{k+n_{a1}} \\ r_{k+n_{a1}+1} \\ \vdots \\ r_{k+n_a} \end{bmatrix}}_{\gamma} + \underbrace{\begin{bmatrix} (\lambda) \\ (\lambda)^2 \\ \vdots \\ (\lambda)^{n_{a1}} \\ 0 \\ 0 \end{bmatrix}}_{W_d} C x_k \tag{7.20}$$

Now, it is convenient to summarise all three suggestions in a suitable form as:

$$\underbrace{\begin{bmatrix} r_{k+1} \\ r_{k+2} \\ \vdots \\ r_{k+n_{a1}} \\ --- \\ r_{k+n_{a1}+1} \\ \vdots \\ r_{k+n_a} \end{bmatrix}}_{\underset{\rightarrow k+1}{r}} = \underbrace{\begin{bmatrix} 1-\lambda^{n_{a1}} & 0 & 0 & \cdots & 0 \\ 1-\lambda^{n_{a1}} & 0 & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1-\lambda^{n_{a1}} & 0 & 0 & \cdots & \vdots \\ -- & -- & -- & -- & -- \\ 0 & I & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & I \end{bmatrix}}_{W_1} \underbrace{\begin{bmatrix} r_{k+n_{a1}} \\ --- \\ r_{k+n_{a1}+1} \\ \vdots \\ r_{k+n_a} \end{bmatrix}}_{\gamma} + \underbrace{\begin{bmatrix} (\lambda)^{n_{a1}} \\ (\lambda)^{n_{a1}} \\ \vdots \\ (\lambda)^{n_{a1}} \\ --- \\ 0 \\ 0 \end{bmatrix}}_{W_2} Cx_k \qquad (7.21)$$

Summary: All three suggestions in the previous subsections can be reduced to the following generic approximation.

$$\underset{\rightarrow k+1}{r} = W_1\gamma + W_2 x_k \qquad (7.22)$$

where γ constitutes the degrees of freedom to encapsulate future values of r_{k+i} and W_1, W_2 are defined appropriately in (7.21). In consequence, the parametric dimension required to include future target information is exactly the dimension of γ.

Now, it is convenient to implement the parametric solution to a dual-mode approach, using approximation (7.22).

7.4 Numerical examples of future target information

In the previous section, we showed how the dimension of parametric space can be reduced using three alternatives. One option is the basic one, in which the future target information is assumed to be full while the other two options are linked to scenarios in which the future target information is reduced. These three alternatives are summarised in one approximation of (7.22). In this section, we present some numerical examples to demonstrate the impact on parametric complexity of including advance knowledge, within the OMPC approach, using the above proposed methods followed by a parametric complexity analysis of the proposed OMPC algorithm for the approximation of (7.22).

First, we consider the following two dimensional state space model and constraints:

$$A = \begin{bmatrix} 0.8 & 0.1 \\ -0.2 & 0.9 \end{bmatrix}; \; B = \begin{bmatrix} 0.3 \\ 0.8 \end{bmatrix};$$

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix} \tag{7.23}$$

$$-0.2 \leq u \leq 0.5; \begin{bmatrix} 1 & 0.2 \\ -0.1 & 0.4 \\ -1 & -0.2 \\ 0.1 & -0.4 \end{bmatrix} x_k \leq \begin{bmatrix} 8 \\ 8 \\ 1.6 \\ 5 \end{bmatrix}$$

The advance knowledge is n_a, the first a few samples is n_{a1} and the degrees of freedom to encapsulate future values of the target is $\gamma = n_a - n_{a1} + 1$ equal to $n_a - n_{a1} + 1$. The parameter λ is set at 0.75, thus, the constant matrices W_1 and W_2 can be easily computed.

The following examples, will focus on the impact of the following parameters choices:

- The advance knowledge is n_a
- The number of the first few samples is n_{a1}
- The degrees of freedom to encapsulate future values of the target is $\gamma = n_a - n_{a1} + 1$.
- The parameter λ and the constant matrices W_1 and W_2.

The parametric solutions are computed with a range of values of n_{a1} and the following information is captured.

- Number of inequalities.
- Number of parametric regions in solution.

7.4.1 Example: 1

For this example, we will use the OMPC Algorithm in Subsection 7.3.2 to demonstrate how the parameters n_a, n_{a1} and γ, influence the parametric complexity. This can be achieved by computing the number of both the regions and the associated inequalities, using a multi-parametric toolbox [78].

Table 7.1 shows the impact of the dimension of the reduced future information (γ) on the parametric solution of the system (7.23) for different values of R, and n_{a1} with $n_a = 3$, using the Algorithm presented in Subsection 7.3.2.

System (7.23) with $n_a = 3$**,** $R = 0.1I, \lambda = 0$			
Dimension of $\gamma = n_a - n_{a1} + 1$	1	2	3
Number of regions	70	85	97
Number of inequalities for all regions	39	32	42
System (7.23) with $n_a = 3$**,** $R = I, \lambda = 0$			
Dimension of $\gamma = n_a - n_{a1} + 1$	1	2	3
Number of regions	91	123	130
Number of inequalities for all regions	30	32	34

Table 7.1: Comparison of parametric solution complexity for different dimensions of γ for the algorithm in Subsection 7.3.2

It is clear from the results that the number of regions and inequalities increase as the dimension of the degrees of freedom (γ) to encapsulate future values of r_{k+i} increases. The effect of R on the parametric solution complexity is clear as, where R increases, the number of regions increases, and the number of inequalities for all of those regions decreases.

7.4.2 Example: 2

For this example, we will use the OMPC Algorithm in Subsection 7.3.3 to demonstrate how the parameters n_a, n_{a1} and γ, influence the parametric complexity.

Table 7.2 shows the impact of the dimension of the reduced future information (γ) on the parametric solution of the system (7.23) for different values of R, and n_{a1} with $n_a = 3$, using the Algorithm in Subsection 7.3.3.

System (7.23) with $n_a = 3$, $\lambda = 0.2$, $R = 0.1I$,			
Dimension of $\gamma = n_a - n_{a1} + 1$	1	2	3
Number of regions	75	106	104
Number of inequalities for all regions	41	41	35
System (7.23) with $n_a = 3$, $\lambda = 0.2, R = I$			
Dimension of $\gamma = n_a - n_{a1} + 1$	1	2	3
Number of regions	91	124	129
Number of inequalities for all regions	31	34	34

Table 7.2: Comparison of parametric solution complexity for different dimensions of γ for the algorithm in Subsection 7.3.3

Similarly, it is clear from the result that the number of both the regions and the inequalities increases as the dimension of the reduced future information (γ) increases. Again, it is notable that, as R increases, the number of regions increases whereas the number of inequalities for all of those regions decreases.

7.4.3 Example: 3: Monte Carlo complexity comparison

Now, in this example, we will demonstrate the impact of including advance knowledge on parametric complexity, within the OMPC approach, using the approximation of (7.22). This can be achieved by making a comparison of the number of regions obtained by implementing the reduced target information scenario with the regions obtained by implementing a full target information scenario.

Our comparison is based on 300 random systems (100 over-damped systems, 100 critically damped systems and 100 open-loop unstable systems) with two states and one input. More detail is shown in appendix A.

All inputs were constrained to:

$$-1 \leq u_k \leq 1.35 \tag{7.24}$$

.

The target limits:

$$-1 \leq r \leq 1 \tag{7.25}$$

The state constraint for all systems are:

$$\begin{bmatrix} 1.6 \\ 5 \end{bmatrix} \leq \begin{bmatrix} 1 & 0.2 \\ -0.1 & 0.4 \end{bmatrix} x_k \leq \begin{bmatrix} 8 \\ 8 \end{bmatrix} \tag{7.26}$$

The tuning parameters are $n_c = 2$, $R = 0.1I$, $Q = C^T C$ and the advance knowledge is chosen as 3. The first few samples to be neglected $n_{a1} = 2$ and the degrees of freedom γ is equal to $n_a - n_{a1} + 1$. λ is set at 0.75 and the parameters W_1 and W_2 can be easily computed from (7.21).

Figures (7.1-7.6) present a comparison of the complexity of the approximation of (7.22) for ($n_{a1} = 1$) and ($n_{a1} = 2$) for 300 random systems. The index of dynamic systems is shown on the x-axis, while the number of regions obtained for each system is shown on the y-axis. The black bar displays the number of regions obtained by using the proposed algorithm with ($n_a = 3, n_{a1} = 1$) or the d.o.f ($\gamma = 3$), while the yellow bar displays the regions obtained from using the proposed algorithm with ($n_a = 3, n_{a1} = 2$) or the d.o.f ($\gamma = 2$).

Figures 7.1 and 7.2 present a comparison of the complexity of the approximation of (7.22) for ($n_{a1} = 1$) and ($n_{a1} = 2$) for 100 over-damped systems with $R = 0.1I$ and $R = 10I$, respectively.

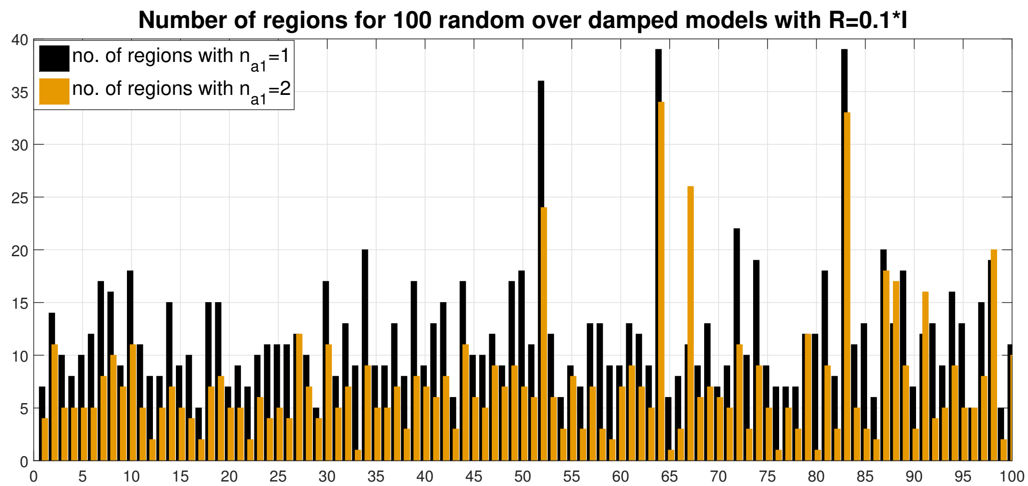

Figure 7.1: Comparison of the approximation of (7.22) complexity with ($n_{a1} = 1$) versus with ($n_{a1} = 2.$) for 100 over-damped systems for $R = 0.1I$

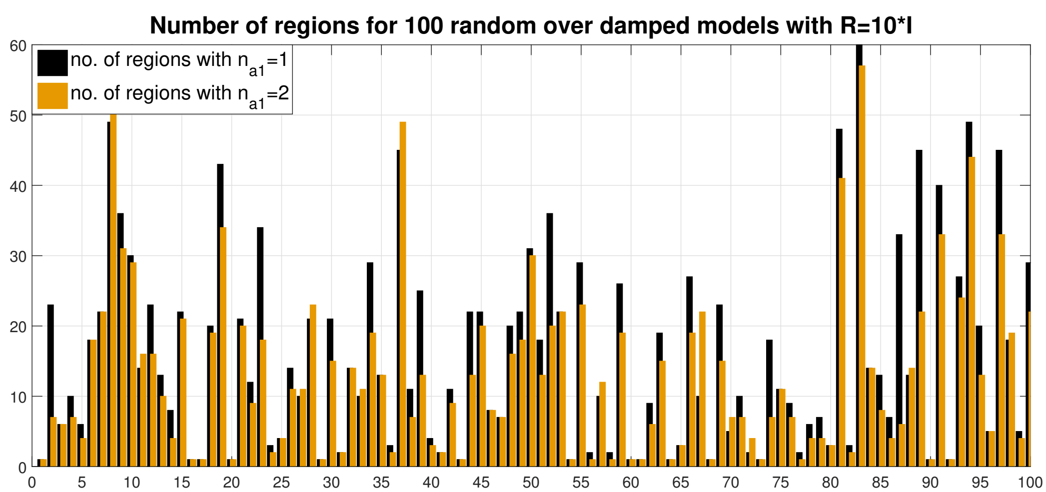

Figure 7.2: Comparison of the approximation of (7.22) complexity with ($n_{a1} = 1$) versus with ($n_{a1} = 2$) for 100 over-damped systems for $R = 10I$

Figures 7.3 and 7.4 present a comparison of the complexity of the approximation of (7.22) for ($n_{a1} = 1$) and ($n_{a1} = 2$) for 100 critically-damped systems with $R = 0.1I$ and $R = 10I$, respectively.

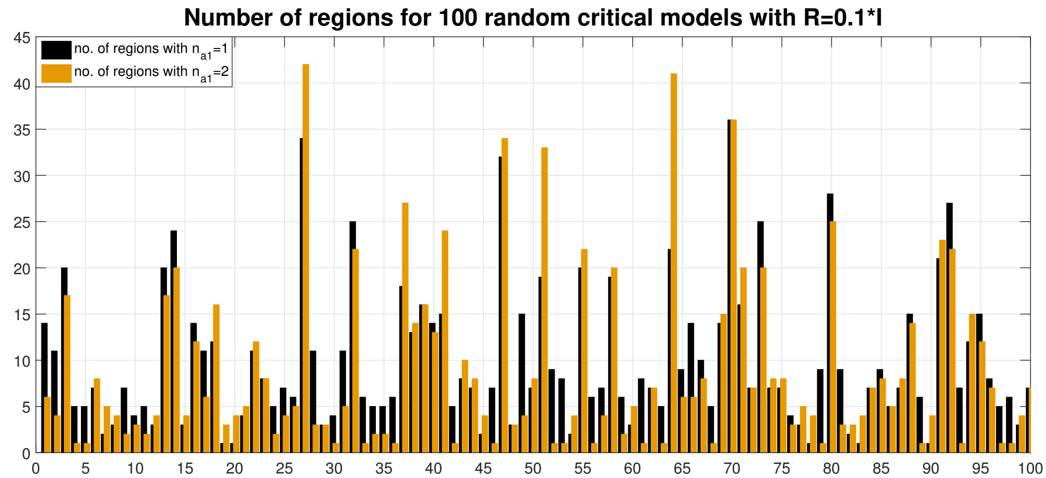

Figure 7.3: Comparison of the approximation of (7.22) complexity with ($n_{a1} = 1$) versus with ($n_{a1} = 2$) for 100 critically-damped systems for $R = 0.1I$.

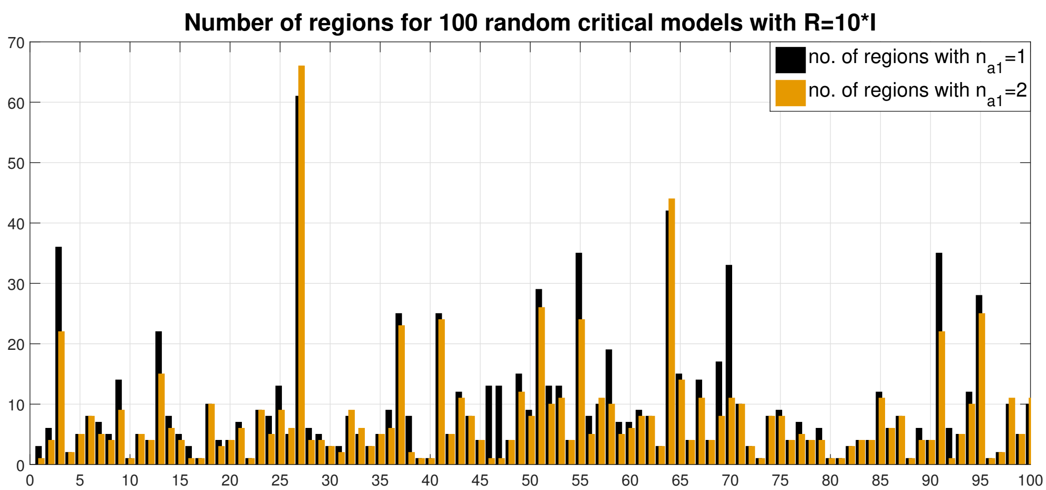

Figure 7.4: Comparison of the approximation of (7.22) complexity with ($n_{a1} = 1$) versus with ($n_{a1} = 2$) for 100 critically-damped systems for $R = 10I$.

Figures 7.5 and 7.6 present a comparison of the complexity of the approximation of (7.22) for ($n_{a1} = 1$) and ($n_{a1} = 2$) for 100 open-loop unstable systems with $R = 0.1I$ and $R = 10I$, respectively.

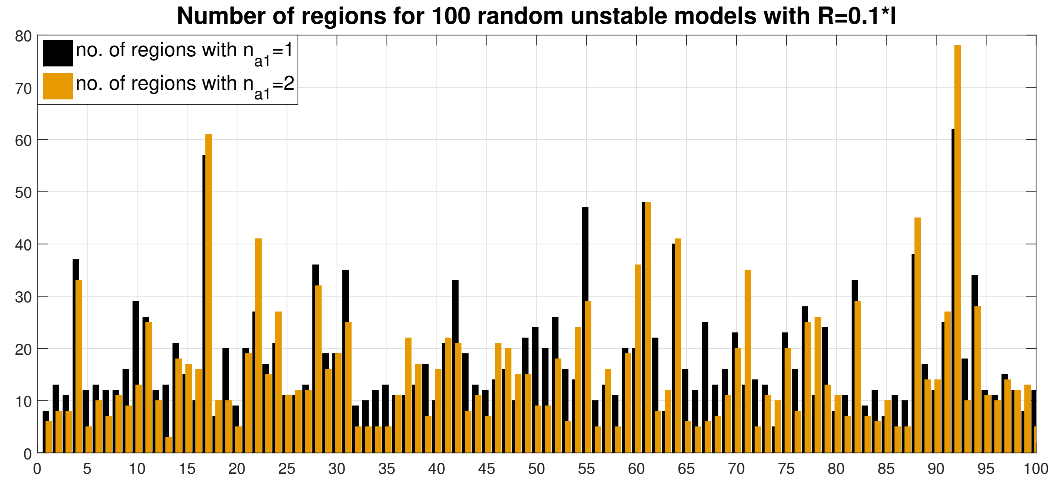

Figure 7.5: Comparison of the approximation of (7.22) complexity with ($n_{a1} = 1$) versus with ($n_{a1} = 2$) for 100 open-loop unstable systems for $R = 0.1I$.

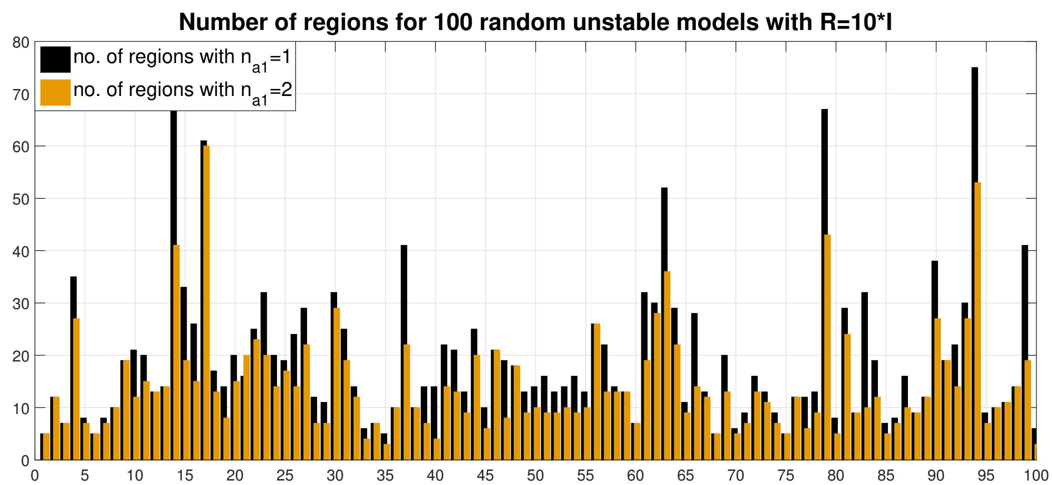

Figure 7.6: Comparison of the approximation of (7.22) complexity with ($n_{a1} = 1$) versus with ($n_{a1} = 2$) for 100 open-loop unstable systems for $R = 10I$.

It can be seen from Figures (7.1-7.6) that the number of regions obtained from using the proposed algorithm with the d.o.f ($\gamma = 2$) is lower than that with the d.o.f ($\gamma = 3$), for most of the systems. It will be noted that the number of regions is affected by the weight, R, where the number of regions (with $R = 10I$) is higher than that in the plots (with

$R = 0.1I$), for both choices of γ, for all of the systems. Moreover, it is noted that the number of regions is also affected by the system dynamics where the open-loop unstable systems relatively offer large number of regions with respect to both the over-damped and critically-damped systems. On the other hand small number of regions are obtained when critically-damped systems are considered.

Summary: It is clear from the results obtained form Figures (7.1-7.6) that the number of regions decreases as the degrees of freedom γ decrease for the majority of systems. This is as expected since the parametric space decreases. It is also noted that, as the weight R varies, the number of regions varies as parametric space as well. It is also noted that, regarding this effect, there is a noticeable difference between any of the systems.

7.5 Feasibility and parametric complexity

The previous sections discussed an issue related to parametric approaches; that is the use of advance information about target changes, as an augmented state in parametric problems. The focus was solely on parametric complexity. In this section, we will discuss a related issue, that is the influence of the input parametrisation on feasibility within OMPC tracking and its parametric solution. We aim to make two contributions. First this investigation reveals that including the core parameters, such as the target and the current input, vastly increases the dimension of the parametric space, with possible consequences for the complexity of any parametric solutions. Secondly, it is shown that a simple re-parametrisation of the d.o.f. can lead to large increases in the feasible volumes, with no increases in the dimension of the required optimisation variables.

7.5.1 Definition of parametric form and consequences

This section summarises some of the of details in the optimal predictive control (OMPC) algorithm discussed in this thesis, paying specific attention to how the problem of equation (7.1) is modified to include the targets and steady-state offsets. Such detail is important for understanding the dimension of the associated parametric space, as extra states are required to include: (i) integral action/targets; (ii) a definition of the input rates; and (iii) the steady-state offsets. The need for each of these states is often tacitly ignored but this increase in dimension has significant repercussions for computational loading and data storage.

7.5.2 Input parametrisation and modified cost function

In this section, we will consider the state space model (7.3), subject to the constraints (7.4).

The future target $\underset{\rightarrow k+1}{r}$ is defined as (assumed constant):

$$\underset{\rightarrow k+1}{r} = [r_{k+1}^T, r_{k+2}^T, \cdots, r_{k+n_y}^T]^T = [I, I, \ldots]^T r_{k+1} \tag{7.27}$$

and is augmented with the current input to the current system state.

The input parametrisation for the case of steady-state offset was described in Section 5.2 as:

$$\begin{aligned} u_{k+i} - u_{ss} &= -K(x_{k+i} - x_{ss}) + c_{k+i} \quad i = 0, 1, \cdots, n_c - 1 \\ u_{k+i} - u_{ss} &= -K(x_{k+i} - x_{ss}) + c_{\infty} \qquad i \geq n_c \end{aligned} \tag{7.28}$$

so the variables $c_{k+i}, i = 0, 1, \ldots, n_c - 1$ are the degrees of freedom which allow deviations in the first n_c moves of the optimal input trajectory; the term c_∞ is a d.o.f. which enables steady-state offset between the asymptotic output predictions and desired target r_{k+1}.

The steady-state states x_{ss} and input u_{ss} were clearly defined in Section 4.1.

An appropriate performance index with the inclusion of steady-state offset was defined in Subsection 5.3.3 as:

$$J = W_1(c_\infty^T S c_\infty) + \underset{\rightarrow k}{c}^T S_c \underset{\rightarrow k}{c} \tag{7.29}$$

where W_1 is a scalar weighting to be selected. Here, the term $c_\infty^T S c_\infty$ penalises asymptotic offset and the term $\underset{\rightarrow k}{c}^T S \underset{\rightarrow k}{c}$ penalises transient performance, including information on $\underset{\rightarrow k+1}{r}$. The scalar weighting W_1 allows users to determine the emphasis they wish to place on each term.

7.5.3 Predictions and the autonomous model

For the implied closed-loop form of (7.28), the predictions must include information such as the future target and current input. A convenient means of combining (7.3, 7.28) is by using an autonomous model formulation, which was discussed in Chapter 4. In this case, the formulation must be extended to capture the evolution of $r_{k+1}, x_k, u_k, \Delta u_k$ as these values appear in the constraints (7.4). Hence, the following identities are needed:

$$\{c_{k+i} = c_\infty, i \geq n_c\} \quad \{r_{k+i} = r_{k+1}, i \geq 1\} \tag{7.30}$$

$$\{\Delta u_{k+i} = u_{k+i} - u_{k+i-1}, i \geq 0\}$$

Combining (7.3,7.28,7.30) gives the following:

$$Z_{k+1} = \underbrace{\begin{bmatrix} \Phi & [B, 0, \ldots, 0] & 0 & 0 & (I-\Phi)K_{xr} \\ 0 & I_L & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ -K & [I, 0, 0, \ldots] & 0 & 0 & K.K_{xr} + K_{ur} \\ 0 & 0 & 0 & 0 & I \end{bmatrix}}_{\Psi} Z_k; \quad Z_k = \begin{bmatrix} x_k \\ \underset{\rightarrow k}{c} \\ c_\infty \\ u_{k-1} \\ r_{k+1} \end{bmatrix}; \tag{7.31}$$

where I_L is a block upper triangular matrix of identities (shift matrix).

7.5.4 The constraints

The final building block in an MPC algorithm is the set of inequalities, which ensure that the predictions from model (7.31) satisfy constraints (7.4). Here,we will use the algorithm discussed in Chapter 5, Section 5.5 to formulate these inequalities. The result is given by following the steps in Section 5.5, showing that all constraints in a standard form $GZ_k \leq f$, as shown in equation (7.32), are suitable for defining the MCAS. The only difference is that the term input rate is included in the augmented state.

$$\underbrace{\begin{bmatrix} -K & [I,0,0,\ldots] & I & 0 & K+K_{xr}+K_{ur} \\ K & -[I,0,0,\ldots] & -I & 0 & -K.K_{xr}-K_{ur} \\ -K & [I,0,0,\ldots] & I & -I & K.K_{xr}+K_{ur} \\ K & -[I,0,0,\ldots] & -I & I & -K.K_{xr}-K_{ur} \\ K_xK_{xr} & 0 & 0 & 0 & 0 \\ -K_xK_{xr} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I \\ 0 & 0 & 0 & 0 & -I \end{bmatrix}}_{G} Z_k \leq \underbrace{\begin{bmatrix} \bar{u} \\ \underline{u} \\ \overline{\Delta u} \\ \underline{\Delta u} \\ \bar{x} \\ \underline{x} \\ \bar{r} \\ \underline{r} \end{bmatrix}}_{f} \tag{7.32}$$

or in a compact form:

$$GZ_k \leq f \tag{7.33}$$

Having the structure of (7.33), one can easily determine the admissible set MCAS, as illustrated in Section 4.5.2, given that:

$$FZ_k \leq d \tag{7.34}$$

the MCAS (7.34), can be described in an expanded form:

$$N \underset{\rightarrow k}{c} + Tc_\infty + Mx_k + Pu_{k-1} + Qr_{k+1} \leq d \tag{7.35}$$

where N, T, M, P, Q are suitable matrices describe the constraints and d is a vector of the limits.

Lemma 7.2 *The constraint inequalities (7.35) can be expressed in parametric form as (d is assumed to be constant):*

$$\underbrace{[N,T]}_{N_T} \begin{bmatrix} \underset{\rightarrow k}{c} \\ c_\infty \end{bmatrix} + \underbrace{[M,P,Q]}_{M_w} w_k \leq d; \quad w_k = \begin{bmatrix} x_k \\ u_{k-1} \\ r_{k+1} \end{bmatrix} \tag{7.36}$$

where the parameter space is w_k and the d.of. are in $\underset{\rightarrow}{c}$, c_∞.

Theorem 7.1 *Including both tracking and input rate constraints into OMPC increases the effective parameter space by the dimensions of the target r_k and the input u_k, respectively, compared to the scenarios where these are excluded.*

Proof: Self-evident from the definition of w_k in (7.36).

Corollary 7.2 *As one aspect of this section is focussed around parametric approaches, future target information has not been included; that is, we assume $r_{k+i} = r_{k+1}, \forall i > 0$. To do otherwise would increase the parametric dimension of w_k still further (Chapter 7).*

7.5.5 The OMPC algorithm with an allowance for steady-state offset

Having constructed all of the foundation components, an OMPC algorithm [153] can be now defined.

Algorithm 7.1 *OMPC is defined as follows. At each sample, perform the quadratic programming optimisation*

$$\min_{\underset{\to k}{c}, c_\infty} \underset{\to k}{c}^T S_c \underset{\to k}{c} + W_1 c_\infty^T S c_\infty \quad \text{s.t.} \quad N_T \begin{bmatrix} \underset{\to k}{c} \\ c_\infty \end{bmatrix} + M_w w_k \leq d; \tag{7.37}$$

Implement the first value of c_k in (7.28) to determine the current input, that is u_k.

Remark 7.2 *Strictly speaking the classical OMPC algorithm uses $c_\infty = 0$, but this section includes the extra d.o.f. because the intention is to consider the efficacy of this for simplifying overall complexity and computational load.*

7.5.6 Summary

This section has defined the core components of an OMPC algorithm which allows for steady-state offset in the predictions, that is an appropriate performance index and also inequalities, to capture the constraints. This offset may be used *optionally* as a mechanism to avoid infeasibility in the transients [51], even where steady-state feasibility is assured. Moreover, the OMPC framework has been cast in a format that is suitable for parametric approaches as these results can now be used to investigate two related but separate issues.

1. The extent to which the parameter c_∞ is more or less effective than $\underset{\rightarrow k}{c}$ in increasing the feasible space.

2. The extent to which the parameter c_∞ may or may not simplify parametric solutions compared to the use of $\underset{\rightarrow k}{c}$ in cases where the problem includes tracking.

7.6 Enlarging the feasible regions using c_∞

In the previous section, we defined a parametric form, which allows for the steady-state offset c_∞ in predictions as an additional d.o.f and includes the targets r_k and current inputs u_{k-1} alongside the current state as a parametric state. In this section, we aim to show how to exploit this steady-state offset in system predictions to obtain large feasible volumes.

The first objective is to assess whether adding the d.o.f. c_∞ is more effective than increasing n_c by one; both of these changes increase the overall d.o.f. and thus the optimisation dimension by the same amount. Any insights gained are useful as, in practice, operators try to keep the overall optimisation dimension as small as reasonably possible.

The concept of n-step sets is widely understood in the MPC literature. In essence:

- A 0-step set is the region in which the control law (7.28) satisfies the constraints when $n_c = 0$ and $c_\infty = 0$. This is where the unconstrained control law is feasible.

- A 1-step set gives the range of values of w_k such that, with a single non-zero value of c_k, one can satisfy the constraints at the first sample, and move into the 0-step set by the next sample.

- A 2-step set gives the range of values of w_k such that, with a single non-zero value of c_k, one can satisfy constraints at the first sample, and move into the 1-step set by the next sample.

- The definition of a n-step set follows the same pattern.

Lemma 7.3 *With $c_\infty = 0$ and a given choice of n_c, the feasible region is given by the n_c-step set. This is obvious.*

Remark 7.3 *Problems occur when the current states x_k, u_{k-1} are at a considerable distance from the target steady-state x_{ss}, u_{ss}. In this case infeasibility can arise as the n_c-step set around the target steady-state is limited in volume, so points far away are not inside it if n_c is small. To retain feasibility, it is necessary to choose an alternative n_c-step set, that is, one associated with a different w_k; this means changing the only component in w_k possible, which is r_k.*

Theorem 7.2 *Where a simple move of the implied steady-state x_{ss}, u_{ss} is sufficient to retain feasibility, then the d.o.f. c_∞ will be sufficient to retain feasibility.*

Proof: This is obvious as choices for c_∞ exist which can be used to imply convergence to any asymptotically stable steady-state point.

Theorem 7.3 *Assuming that one was feasible at sample $k-1$, then the inclusion of c_∞ guarantees feasibility at the current sample.*

Proof: The main difference between sample k and $k-1$ in terms of the implied predictions in (7.36) is the change in the value r_{k+1}. It has been shown that c_∞ can overwrite any impact on predictions from a change in that state, and thus can be used to place the system in the same effective state as was the case in the previous sample.

Corollary 7.3 *It is noted that reachable steady-states are limited to the sub-space implicit in (7.6). Where a simple move of the implied steady-state x_{ss}, u_{ss} is not sufficient to retain feasibility, then the d.o.f. c_∞ is less likely to be useful and hence one obtains more benefit from increasing n_c. This case will occur where the initial condition rather than changes in the target cause infeasibility.*

7.7 Numerical examples of feasibility

This section will show how the shapes of the feasible regions vary for changes in n_c with the inclusion or not of c_∞. We will consider two cases; one is where the current input varies while the other case is where the target varies.

7.7.1 Feasibility with and without c_∞ but varying u_{k-1}

A little discussed issue in the literature is the impact of the initial input on the feasible regions; this is relevant when there exist input rate constraints and it is also clear that u_{k-1} is one component of the parametric space w_k. This subsection shows how the feasible region's shape and volume change substantially as the current input changes. For ease of illustration, the examples are restricted to a parameter space of dimension two.

Example: 1

Consider the following state space model:

$$A = \begin{bmatrix} 0.8 & 0.1 \\ -0.2 & 0.9 \end{bmatrix}; \; B = \begin{bmatrix} 0.3 \\ 0.8 \end{bmatrix}; \tag{7.38}$$

subject to the input, input rate and state constraints respectively:

$$-0.2 \le u \le 0.5; \quad \|\Delta u_k\| \le 0.05; \begin{bmatrix} 1 & 0.2 \\ -0.1 & 0.4 \\ -1 & 0.2 \\ 0.1 & -0.4 \end{bmatrix} x_k \le \begin{bmatrix} 8 \\ 8 \\ 1.6 \\ 5 \end{bmatrix}$$

Figure 7.7 shows how the 2-step set for system (7.38) changes as u_{k-1} changes for a standard OMPC algorithm without c_∞ while Figure 7.8 shows how the 1-step set changes for system (7.38) as u_{k-1} changes but also including the d.o.f. c_∞.

It can be seen from Figures 7.7 and 7.8 that, when the steady state offset is included

($c_\infty \neq 0$), the obtained feasible regions for various current inputs offer larger volumes than those obtained when no steady state offset ($c_\infty = 0$) is considered.

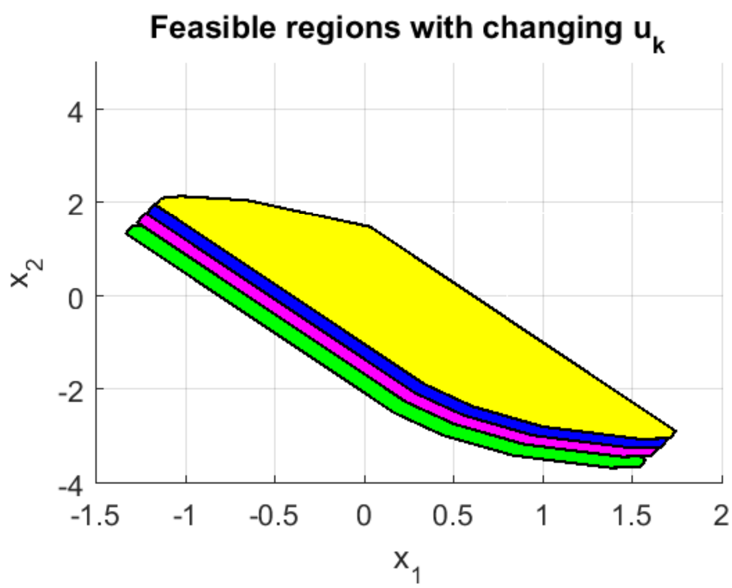

Figure 7.7: Variation in the feasible region of system (7.38) with $n_c = 2, r_{k+1} = 0$ and $u_{k-1} = 0.5, 0.2, 0, -0.2$.

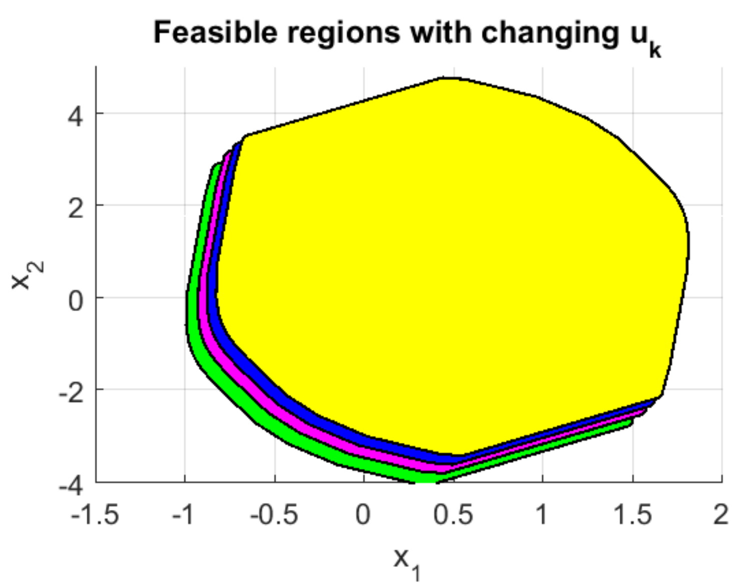

Figure 7.8: Variation in the feasible region of system (7.38) with $n_c = 1, r_{k+1} = 0$ and $c_\infty \neq 0$ and $u_{k-1} = 0.5, 0.2, 0, -0.2$.

Example: 2

Consider the following state space model:

$$A = \begin{bmatrix} 0.8 & -0.53 \\ -0.09 & 0.97 \end{bmatrix}; \; B = \begin{bmatrix} 0.09 \\ 0.005 \end{bmatrix}; \qquad (7.39)$$

subject to the input, input rate and state constraints, respectively:

$$-5 \leq u \leq 4; \quad \|\Delta u_k\| \leq 0.4; \begin{bmatrix} 1 & 0.2 \\ -0.1 & 0.4 \\ -1 & -0.2 \\ 0.1 & -0.4 \\ -1 & -0.45 \end{bmatrix} x_k \leq \begin{bmatrix} 4 \\ 1.6 \\ 0.8 \\ 1.6 \\ 0.6 \end{bmatrix} \qquad (7.40)$$

Figures 7.9 shows how the 2-step set, for system (7.39), changes as u_{k-1} changes for a standard OMPC algorithm without c_∞ and Figures 7.10 show how the 1-step set changes for the same example as u_{k-1} changes but also including the d.o.f. c_∞.

Again, it can be seen from Figures 7.9 and 7.10 that the feasible regions which are obtained for various current inputs for the case when $c_\infty \neq 0$ offer larger volumes than those for the case when $c_\infty = 0$.

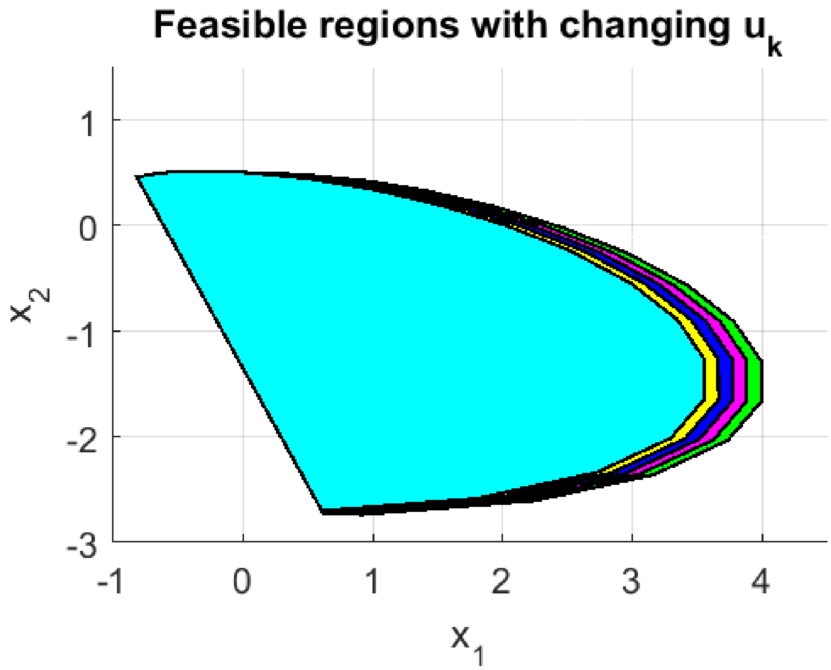

Figure 7.9: Variation in the feasible region of system (7.39) with $n_c = 2, r_{k+1} = 0$ and $u_{k-1} = 2, 1, 0, -1, -2$.

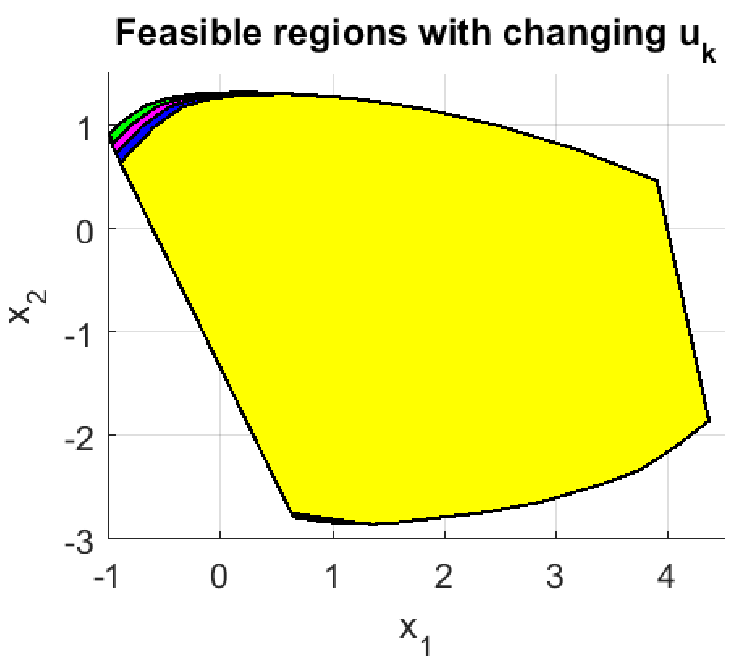

Figure 7.10: Variation in the feasible region of system (7.39) with $n_c = 1, r_{k+1} = 0$ and $c_\infty \neq 0$ and $u_{k-1} = 2, 1, 0, -1, -2$.

Two conclusions are obvious: (i) First it is essential that u_{k-1} is included as a parametric state and, moreover, this can have a significant impact on whether a given x_k is feasible or not;(ii) Secondly, in this case, by adding a d.o.f. c_∞ as opposed to c_{k+1} as given significant enlargements in the feasible region, Figures (7.7-7.8) and (7.9-7.10) have the same number of d.o.f. but clearly the latter of each pair has a larger volume.

7.7.2 Feasibility with and without c_∞ but varying r_k

The literature has tended to focus on feasible regions where the concern is the initial condition and regulation, with an almost tacit assumption that the target is the origin. In practice, the target may change and this can have significant effects on the shape of the feasible region. In such a case, the traditional OMPC d.o.f., that is $\underset{\rightarrow k}{c}$ may, or may not, be effective.

This subsection uses example 1 and shows how the feasible region's shape and volume changes substantially as the target changes and moreover emphasises that the standard d.o.f. in $\underset{\rightarrow k}{c}$ may have a limited impact in dealing with this.

Figure 7.11 shows how the 2-step set changes as r_{k+1} changes for a standard OMPC algorithm without c_∞ and Figure 7.12 shows how the 1-step set changes as r_{k+1} changes but also including the d.o.f. c_∞.

It is clear form Figure 7.12 that the algorithm which includes c_∞, has a feasible region which is completely unaffected by the choice of r_{k+1}. It will be noted that equivalent feasible regions are obtained for all choices of the target since these targets are within equivalent degrees of freedom c_∞. In retrospect, this is to be expected but of course it demonstrates the huge benefit of this option as opposed to the conventional OMPC algorithm whose feasible regions, as shown in Figure 7.11, are far smaller in comparison, with the inevitable risk that frequent infeasible scenarios may arise. The most significant point here is that, if c_∞ is included as a d.o.f., the feasible region is unaffected by the choice of r.

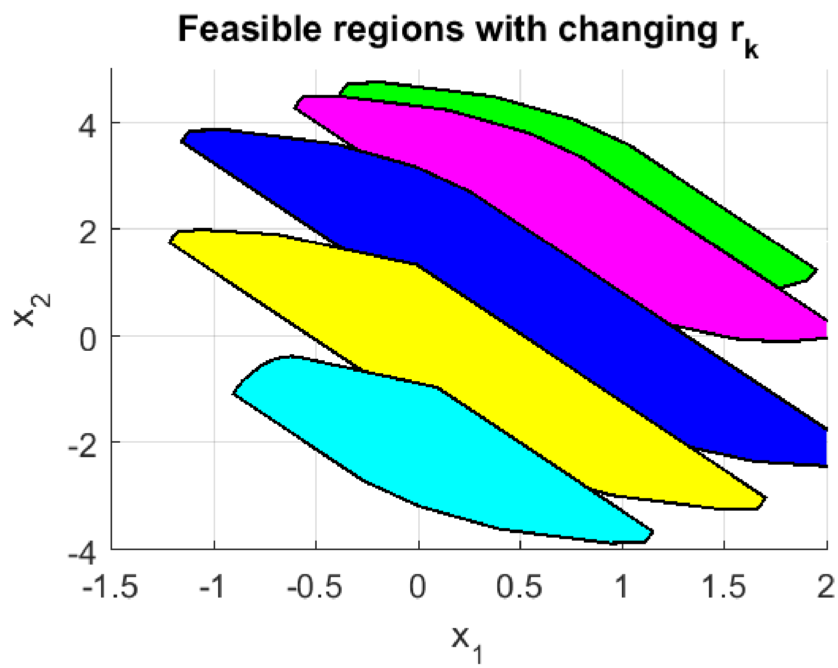

Figure 7.11: Variation in the feasible region of system (7.38) with $n_c = 2, u_{k-1} = 0$ and $r_{k+1} = -1, -0.5, 0, 0.5, 1, 1.2$.

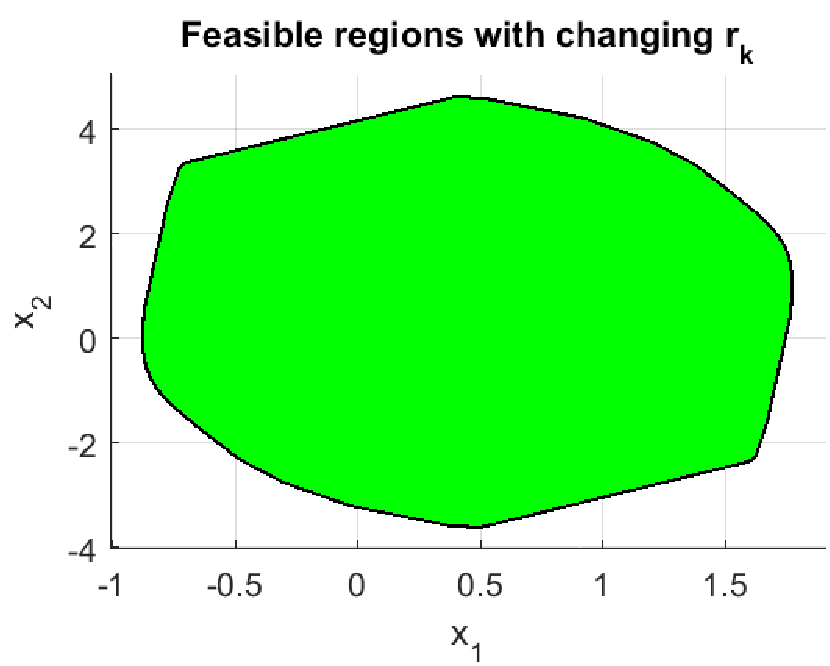

Figure 7.12: Variation in the feasible region of system (7.38) with $n_c = 1, u_{k-1} = 0$ and $c_\infty \neq 0$ and $r_{k+1} = -1, -0.5, 0, 0.5, 1, 1.2$.

7.8 Simplifying parametric solutions complexity using c_∞

In the previous section, we have shown potential feasibility benefits of exploiting the steady-state offset in a closed-loop prediction paradigm, another interesting point is to wonder if this steady-state offset (d.o.f), can be used to simplify the complexity of a parametric solution. That is, if one can obtain a similar volume feasible region with far fewer d.o.f., is it possible that one may also require far fewer parametric regions?

At this point, it is interesting to perform some case studies to find the complexity of the associated parametric solution. Where c_∞ is included, the implied number of d.o.f. is one higher and of course the volumes of the feasible regions differ, but here the focus is solely on the parametric solution complexity.

This section presents the results of simplifying the parametric solution complexity,using the algorithm of [78], as shown in Tables 7.3 and 7.4, and also using the same examples as in the previous section; it is accepted this is a very narrow snapshot and a far broader investigation is possible.

Number d.o.f.	2	3	4	5	6
Without c_∞,$r_k = 0, u_{k-1} = 0$	32	58	79	105	142
With c_∞, $r_k = 0, u_{k-1} = 0$	32	58	79	105	189
Without c_∞, $r_k = 1, u_{k-1} = 0.5$	17	37	55	79	118
With c_∞, $r_k = 1, u_{k-1} = 0.5$	25	51	80	108	218

Table 7.3: Comparison of the number of regions in the mp-QP solution with a d.o.f. of just c_k and with (c_k, c_∞) for system (7.38).

Summary: There is no obvious pattern, but of course one could argue that including c_∞ gives much larger feasible volumes in general for the same number of d.o.f. so, for equivalent volumes of feasible regions, it is likely that using c_∞ will result in far fewer parametric regions.

Number d.o.f.	2	3	4	5	6
Without c_∞,$r_k = 0, u_{k-1} = 0$	40	86	138	193	255
With c_∞, $r_k = 0, u_{k-1} = 0$	40	86	138	193	355
Without c_∞, $r_k = 1, u_{k-1} = 2$	22	59	102	151	218
With c_∞, $r_k = 1, u_{k-1} = 2$	16	44	72	104	237

Table 7.4: Comparison of the number of regions in the mp-QP solution with a d.o.f. of just c_k and with (c_k, c_∞) for system (7.39).

7.9 Conclusions

Foremost, it is clear that including advance information about targets increases the dimension of the parameter space for a parametric approach to predictive control. It is recognised that a parametric solution is often impractical for large parameter spaces and thus one may infer that, usually, a parametric approach would be difficult to use in conjunction with advance knowledge scenarios. Nevertheless, this chapter has introduced some reformulations of a typical dual-mode (OMPC) algorithm which can, to a limited extent, overcome problems with dimension growth. It might be helpful to exploit the 'added value' in the future target information and capture this value in fewer variables; in essence, the increase in the parameter space is linked to the number of variables needed to capture the useful information in the target trajectory and, if needed, one can capture this with very few variables and thus reduce the dimension to that required for incorporating the optimisation variables. Clearly, any simplification of the target information results in some degree of sub optimality, but that is likely to be a price worth paying to improve in the simplicity of the parametric solution.

In general, reducing the target information dimension should simplify the parametric solution, but here it is shown that, while the result obtained was as we expected for most systems, this is not true for all systems. Moreover, the weight R may influence the number of regions and indeed many other factors, such as the model or constants. Therefore, we can argue that it is difficult to obtain analytical results and impossible to apply this concept for a specific system. In general, this best expectation is not guaranteed.

In summary, the current parametric approaches are unsuitable for exploiting future target information in general.

This chapter also investigates a feasibility issue in predictive control. It is known that the volume of the feasible regions is linked to the number of d.o.f., and indeed the choice of terminal control law, but the consequences of the target and the current input, indeed the visibility of the importance of the current value of the input, is a core contribution. It is shown here that the systematic inclusion of steady-state offset allows potentially substantial increases in feasible volumes and thus caters for a number of important scenarios which otherwise could lead to infeasibility. The benefits cannot be proven in general, beyond the obvious scenario of set point changes, and will vary from case to case.

Chapter 8

CASE STUDIES

All proposed OMPC feed-forward algorithms were successfully implemented in the previous chapters using different process dynamics by means of their mathematical models. These algorithms have demonstrated the systematic use of future information about target or disturbance changes for tracking both reachable and unreachable time-varying targets. The aim is to design an efficient feed-forward for both the certain (nominal) and uncertain (robust) cases. Moreover, the algorithms demonstrated the feasibility assurance and stability guarantee for time-varying targets. In order to examine the applicability of the proposed algorithms to the industry, we will demonstrate the advantages of the proposed algorithms in this chapter with regard to processes undertaken within different industries, such as aeronautical, petrochemical and industrial processes. This chapter starts with a description of the basic components required for the proposed OMPC feed-forward design for both the nominal and the robust design. Section 8.2 presents the implementation of the proposed feed-forward design in a fighter aircraft as a nominal case. Section 8.3 presents a nominal feed-forward design for an upstream gas process while Section 8.4 presents the robust design for a mechanical process with parametric uncertainty. The chapter ends by offering a discussion and conclusions.

8.1 Background components

In this section, we will explore the basic components required for these case studies as follows.

8.1.1 Preview (advance knowledge) with predictive control

Advance information on the target to be tracked is commonly available in most scenarios. One example of this is car driving. The driver needs advance information about the path ahead, such as corners, hills, or pedestrians crossing the road. This enables the driver to make decisions earlier to avoid any expected risks arising while driving. Another example is a climate regulation system, which makes advance information about solar radiation available and to enables the controller to take action regarding the desired temperature and humidity regulation. This phenomenon can be translated into a predictive control approach by taking into account the future information about changes in the input references when good reference tracking performance is required.

In conjunction with our contribution in this thesis, we aim to apply the OMPC algorithms proposed in this thesis to the processes where advance information about target changes is available and its inclusion is very important. One example of the availability of advance information about the target or reference changes is found in a flight control system. An autopilot system requires advance information about the target changes, from the pilot, in order to prepare for the control action needed to reach the desired condition during the flight. Another example can be found in refinery plant such as oil or gas refineries, where the controller should be prepared to take the control action prior to any reference changes in order to achieve the desired performance. The application of the proposed algorithms to such examples are presented in the following sections.

8.1.2 The nominal OMPC feed-forward algorithm

We will describe in this subsection, the basic components of the proposed OMPC feed-forward control nominal design, which were presented in Section 5.5, as follows. The corresponding performance index, J, is given by:

$$J_p = W_1(c_\infty^T S c_\infty) + \underset{\rightarrow k}{\tilde{c}}^T S_c \underset{\rightarrow k}{\tilde{c}} \tag{8.1}$$

where W_1 is a scalar weighting to be selected. Here, the term $c_\infty^T S c_\infty$ penalises asymptotic offset and the term $\underset{\rightarrow k}{\tilde{c}}^T S_c \underset{\rightarrow k}{\tilde{c}}$ penalises transient performance, including information on $\underset{\rightarrow k+1}{r}$.

The scalar weighting W_1 allows the user to determine the emphasis that he/she wishes to place on each term.

The input and state or output constraints that are applied for the process to be controlled can be described in a standard form as:

$$S_{MCASU} = \{x : \exists(\tilde{c}_k, c_\infty) \;\; s.t \; Mx_k + N \underset{\rightarrow k}{\tilde{c}} + Tc_\infty + Q \underset{\rightarrow k+1}{r} \leq t\} \tag{8.2}$$

where $M,\ N,\ T$ and Q are suitable matrices and t is a vector of the limits.

Thus, it is straightforward to apply Algorithm 5.1 as follows.

$$\min_{\underset{\rightarrow}{\tilde{c}},\ c_\infty} \quad W_1(c_\infty^T S c_\infty) + \underset{\rightarrow k}{\tilde{c}}^T S_c \underset{\rightarrow k}{\tilde{c}} \qquad s.t. \qquad Mx_k + N\tilde{c}_k + Tc_\infty + Qr_{k+1} \leq t \tag{8.3}$$

We use the optimised $\underset{\rightarrow}{\tilde{c}},\ c_\infty$ in conjunction with (5.6) and (5.13) to determine c_k and implement the first move u_k of the control law, as defined in (5.1).

8.1.3 The robust OMPC feed-forward algorithm

In order to use the robust feed-forward design, we will present the basic components for the proposed OMPC feed-forward control robust design, which was presented in Subsection 6.5.4, as follows. The corresponding performance index, J, is given by:

$$J = W_1(c_\infty^T S c_\infty) + \underset{\rightarrow k}{c}^T S_c \underset{\rightarrow k}{c} \tag{8.4}$$

The proposed robust MCAS (6.64) was also defined in Chapter 6 as:

$$M_r x_k + N_r \underset{\rightarrow k}{\tilde{c}} + Q_r c_\infty + V_r \underset{\rightarrow k+1}{r} \leq d_r \tag{8.5}$$

where $M_r,\ N_r,\ Q_r$ and V_r are suitable matrices and d_r is a vector of the limits.

Thus, it is straightforward to define Algorithm 6.3, which will be used in this section.

$$\min_{c_\infty, \tilde{c}_k} \quad J \quad s.t. \;\; M_r x_k + N_r \underset{\rightarrow k}{\tilde{c}} + Q_r c_\infty + V_r \underset{\rightarrow k+1}{r} \leq d_r \tag{8.6}$$

We implement the first block element of $\tilde{c}_k$ in (6.48) to compute the control law as defined in (6.46).

8.1.4 Steady state calculation and constraints

For a given model dynamics with a desired target, one can compute the steady-state and inputs, x_{ss}/y_{ss}, u_{ss}, with $x_{ss} = k_{xr} * r_k$ and $u_{ss} = k_{ur} * r_k$, respectively, as defined in Chapter 4, equation (4.3). Therefore, one can apply limits on both the input and states/outputs, such that the target becomes feasible. This can be achieved if the steady-state values lie within the applied constraints. We will use this concept in the following studies.

8.2 Fighter aircraft manoeuvre limiting using a feed-forward nominal design

One important issue in fighter flight control is the so-called flight manoeuvre limiting. This is defined as the ability to restrict the response of the aircraft to the pilot input such that the states remain within a flight envelope [159]. A systematic way of incorporating manoeuvre limiting in the fighter aircraft control system is to add a restriction to both the angle of attack and the pitch rate as well as the pilot input stick command [159]. Therefore, the main goal of the control design in fighter aircraft is to track the angle of attack while keeping it within a specific range in order to avoid any risky conditions.

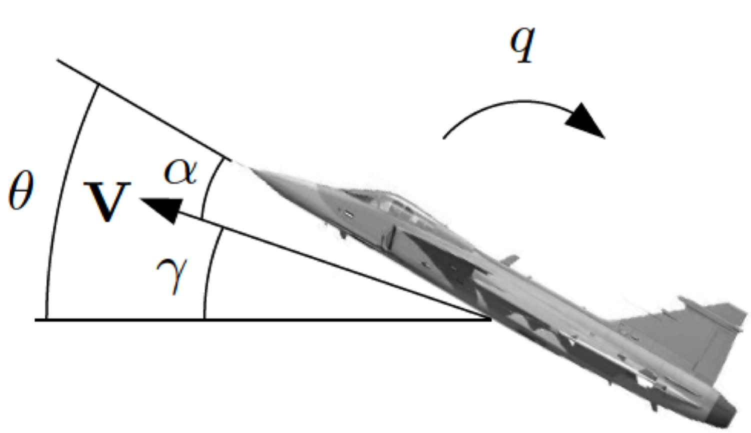

Figure 8.1: Definition of angles for aircraft control [60]

In this section, we will implement the feed-forward design algorithm described in Section 8.1.2 to a fighter aircraft [97] to fulfil manoeuvre limiting. Subsection 8.2.1 presents the model description of the gas treatment model. Subsection 8.2.2 demonstrates the nominal design for reachable targets while Subsection 8.2.3 examines the nominal design for unreachable targets.

8.2.1 The aircraft model description

In this study, we will use the model of the fighter aircraft that has been used in [159]. The aircraft angles are defined in Figure 8.1. The pilot will fly the aircraft under a specified desired flight condition. The manipulated variable is the stick command, δ_k and one of the command outputs that must be controlled is the angle of attack, α_k. The critical states that must be restricted are the angle of attack, α_k and the pitch rate, q_k [159]. The discrete state space model of the aircraft is described in [159] for sampling period, $T_s = 60\ ms$, as follows. The state equation is given by:

$$\begin{bmatrix} \alpha_{k+1} \\ q_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} 0.9719 & 0.0155 \\ 0.2097 & 0.9705 \end{bmatrix}}_{A} \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix} + \underbrace{\begin{bmatrix} 0.0071 \\ 0.3263 \end{bmatrix}}_{B} \delta_k \tag{8.7}$$

The output equation is given by:

$$y_k = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix} \tag{8.8}$$

where $\alpha_k,\ q_k$ are the states, δ_k is the input command and y_k is the output.

The goal of the design is to track a desired reference angle of attack for the aircraft while keeping the states of the aircraft dynamics that are corresponding to the angle of attack and the pitch rate as well as the stick input command within a specific range. In this study, we will demonstrate the benefits of using advance knowledge on reference future changes in the feed-forward design for the nominal case. Moreover, we will show how simply the proposed algorithm can handle constraints in the control design for both reachable and unreachable targets, which provides better reference tracking performance than is the case with no advance knowledge.

8.2.2 Assuming the target is reachable at s steady-state

For the corresponding aircraft dynamics, the steady-state values u_{ss} and x_{ss} can be computed with $x_{ss} = k_{xr} * r_k$ and $u_{ss} = k_{ur} * r_k$ as follows. The values of k_{xr} and k_{ur} can be computed by using formula (2.8) in Chapter 2 as:

$$\begin{bmatrix} k_{xr} \\ k_{ur} \end{bmatrix} = \begin{bmatrix} C & 0 \\ A - I & B \end{bmatrix}^{-1}; \quad \Longrightarrow \quad k_{xr} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \quad and \quad k_{ur} = \begin{bmatrix} -0.4597 \end{bmatrix} \tag{8.9}$$

Let us assume that the aim is to track a reference angle of attack (target) of $r_k = 28$; hence, the steady-state values can be defined as:

$$x_{ss} = k_{xr} * r_k = \begin{bmatrix} 28 \\ 56 \end{bmatrix}; \quad u_{ss} = k_{ur} * r_k = \begin{bmatrix} -12.9 \end{bmatrix} \tag{8.10}$$

In order to demonstrate how proposed Algorithm 5.1 can handle the reachable targets, we assume that the reference input to be tracked (target) of $r_k = 28$ and apply constraints on the angle of attack, α_k, the pitch rate, q_k and the stick input command, δ_k, such that the target is feasible at a steady-state as:

$$\begin{bmatrix} -10 \\ -80 \end{bmatrix} \leq \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix} \leq \begin{bmatrix} 28 \\ 80 \end{bmatrix}; \quad \begin{bmatrix} -20 \end{bmatrix} \leq \delta_k \leq \begin{bmatrix} 36 \end{bmatrix} \tag{8.11}$$

With these state constraints (8.11), the target can be asymptotically reached (feasible) since the upper limit of the angle of attack is equal to its corresponding steady-state value in equation (8.10), the upper limit of the pitch rate exceeds its corresponding steady-state value in equation (8.10) and the lower limit on δ_k exceeds its corresponding absolute steady-state value of equation (8.10), as discussed in the previous subsection. This ensures that the output can asymptotically converge to its steady-state value.

Next, we will perform a closed-loop simulation for the system (8.7) subject to the applied constraints (8.11), with and without advance knowledge of the reference changes. The tuning parameters are as follows: the control horizon is $n_c = 3$, the state weighting matrix is $Q = C^T C$ and the weight in the input command is $R = 0.1I$. As discussed in Chapter

4, when no advance knowledge is considered the value of n_a is set to 1 but when the advance knowledge is considered, the value of n_a can be appropriately chosen by using the formulation of Algorithm 4.1, introduced in Chapter 4. In this case, the appropriate value of $n_a = 10$, can be useful. This will be demonstrated in the following simulation.

Figure 8.2 shows the closed-loop responses of the angle of attack tracking for the system (8.7) with $n_a = 1$ and $n_a = 10$.

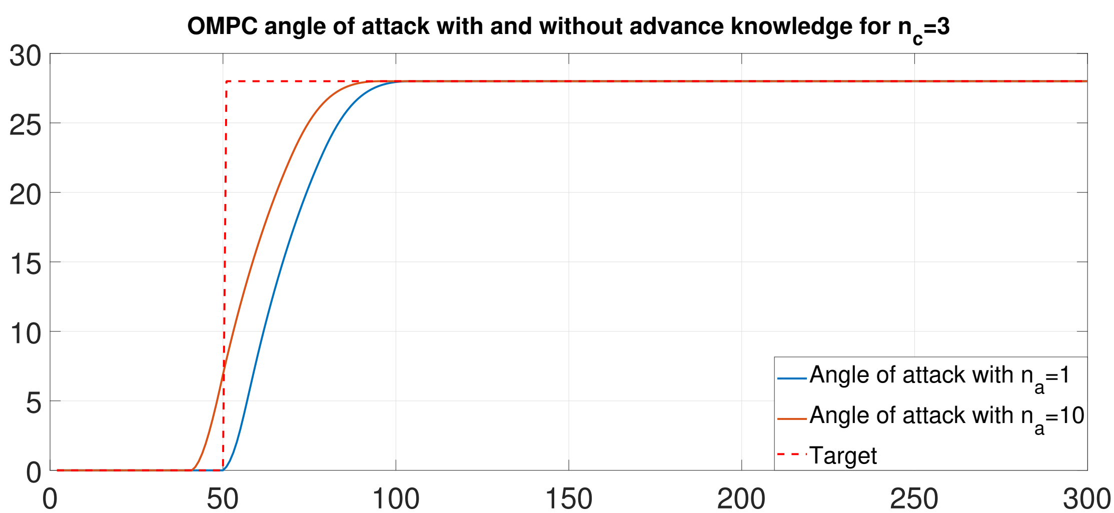

Figure 8.2: Closed-loop for the step responses of system (8.7) for the reachable target with $n_a = 1$ and $n_a = 10$

It is clear to see that the aircraft angle of attack response with advance knowledge of $n_a = 10$ is faster than that without advance knowledge $n_a = 1$, in which the settling time with $n_a = 1$ is about 95 sample time, while for $n_a = 10$, about 85 sample time. This implies that the response improves when advance information on the forward speed reference is considered.

Figure 8.3 shows the evolution of the pilot input commands for the angle of attack tracking tracking with and without advance information about the reference changes.

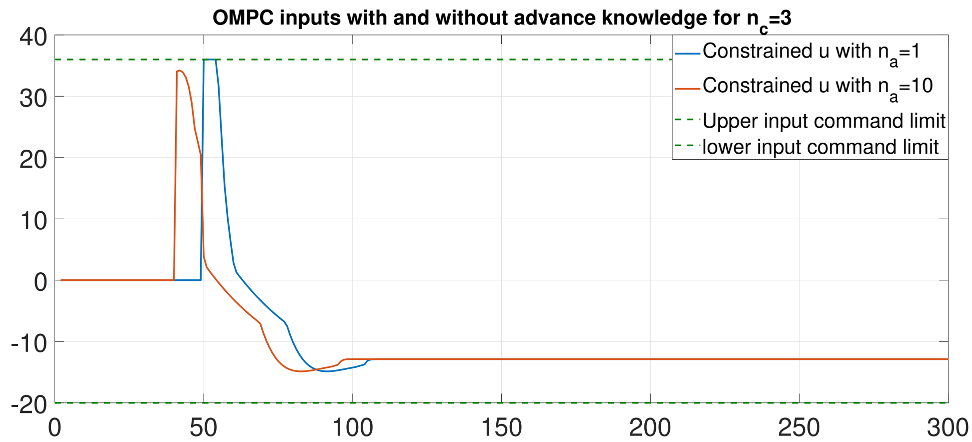

Figure 8.3: the evolution of the pilot input command for the angle of attack tracking tracking for a reachable target with $n_a = 1$ and with $n_a = 10$

It can be observed that when advance knowledge is considered ($n_a = 10$), the initial control effort level is slightly lower than is the case when no advance knowledge ($n_a = 1$) is considered. The key observation here is that the input constraints are not active when appropriate advance knowledge is considered, while they are active when no advance knowledge is considered.

Figure 8.4 shows the evolution of the perturbations about optimal at each sample for the aircraft angle of attack, with and without advance information about the reference changes.

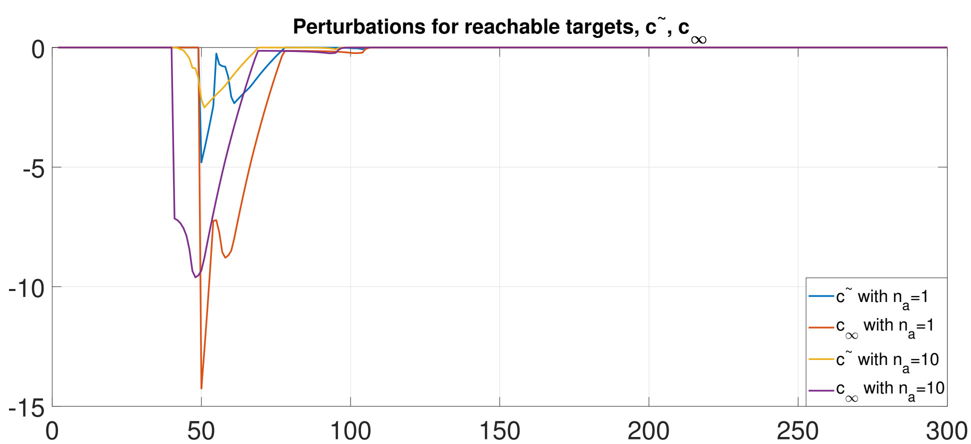

Figure 8.4: the evolution of the perturbations about optimal for the reachable target with $n_a = 1$ and with $n_a = 10$

Readers will notice that the perturbations about the optimal $\tilde{c_k}$, are non-zeros at the time of the set point change since the constraints are active during that time. They will also notice that, c_∞ is non-zero during transients but becomes zero at the steady-state. This is because the target is infeasible only at that time and becomes asymptotically feasible at the steady-state. Regarding the inclusion of future information about target changes, it is observed that, when advance knowledge is considered, the degree of perturbations is lower than is the case when no advance knowledge is available. Moreover, the perturbations with advance knowledge become zero earlier than is the case when no advance knowledge is available.

8.2.3 Assuming the target is unreachable

In this subsection, we will show how our proposed algorithm can effectively handle unreachable targets with and without the inclusion of advance knowledge in the optimisation.

Once again, suppose the aim is to track an aircraft angle of attack of $r_k = 28$ with applied constraints on both the angle of attack and the pitch rate, α_k and q_k, respectively as follows.

$$\begin{bmatrix} -10 \\ -55 \end{bmatrix} \leq \begin{bmatrix} \alpha_k \\ q_k \end{bmatrix} \leq \begin{bmatrix} 28 \\ 55 \end{bmatrix} \tag{8.12}$$

In this case the target cannot be reached in either the transient or the steady-state since the pitch rate of $q_k = 55$ is too small to provide the steady-state value in equation (8.10). Now, we can apply limits on the inputs such that they exceed their corresponding steady-state values of equation (8.10).

$$-25 \leq \delta_k \leq 25 \tag{8.13}$$

A closed-loop simulation for the system (8.7) is performed subject to the applied constraints (8.13) and (8.12), with and without advance knowledge of the reference changes. The tuning parameters are; the control horizon is $n_c = 3$, the weighting matrix is $Q = C^T C$ and the input weight is $R = 0.1I$. The advance knowledge can be set to $n_a = 10$, as discussed in the previous subsection.

Figure 8.5 shows the closed-loop angle of attack responses for the system (8.7) with $n_a = 1$ and with $n_a = 10$.

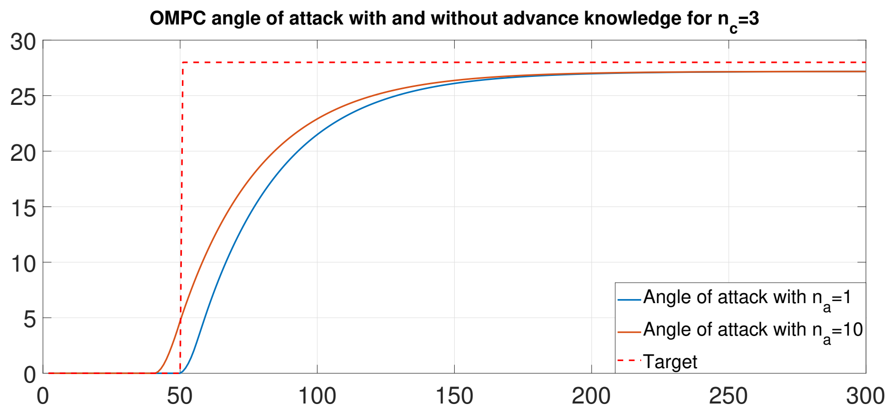

Figure 8.5: Closed-loop for angle of attack step responses of system (8.7) for an unreachable target with $n_a = 1$ and with $n_a = 10$

It can be seen that the output response with advance knowledge is faster than without advance knowledge. Moreover, the output cannot reach the target but converges to the nearest point to the target since the state variable (pitch rate q_k) is lower than its corresponding steady-state value.

Figure 8.6 shows the evolution of the pilot input commands for the aircraft forward speed tracking with and without advance information about the reference changes.

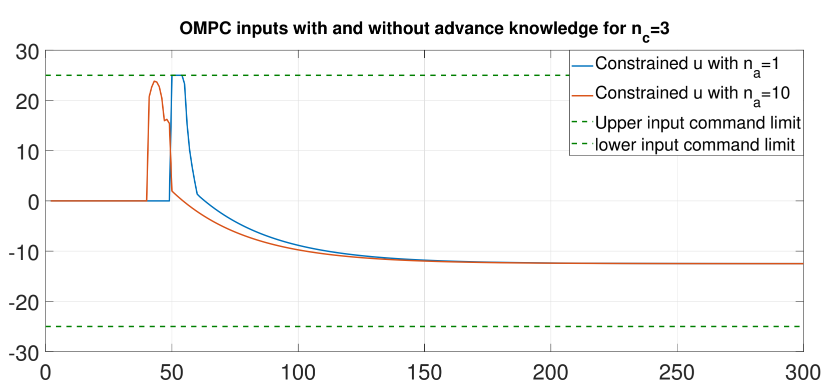

Figure 8.6: the evolution of the pilot input commands for the aircraft forward speed tracking for an unreachable target with $n_a = 1$ and with $n_a = 10$

It can be observed that, with effective use of advance knowledge ($n_a = 10$), less control effort is required than is the case with no advance knowledge ($n_a = 1$). It is also observed that the input constraints are not active when advance knowledge is considered while they are active when no advance knowledge is considered.

Figure 8.7 shows the evolution of the perturbations about optimal at each sample for the aircraft forward speed tracking with and without advance information about the reference changes.

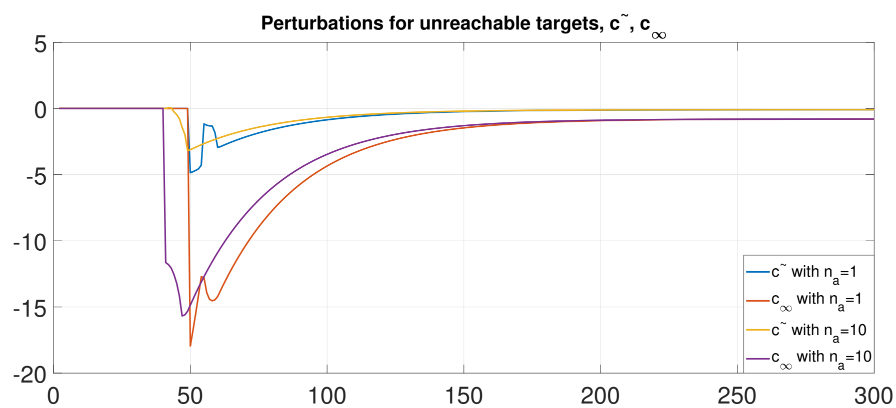

Figure 8.7: the evolution of the perturbations a bout optimal for unreachable target with $n_a = 1$ and with $n_a = 10$

It can be clearly seen that the perturbations about the optimal c_∞ are non-zero at the time of the set point change as well as at the steady-state since the target is infeasible during both the transient and steady-state. It can also be seen that the perturbations about the optimal $\tilde{c_k}$ are non-zero only at the time of the set point change since the input constraints are active at that time. It is observed that when advance knowledge is considered, the degree of perturbations is lower than is the case with no advance knowledge. Moreover, the perturbation term c_∞ with advance knowledge reaches its steady-state offset earlier than is the case with no advance knowledge. To seek further evidence of the benefits of the effective use of advance knowledge in the feed-forward design, we will make a comparison between the performance indices J for the system (8.7) with and without advance knowledge, as shown in Table 8.1.

	J with $n_a = 1$	J with $n_a = 10$
System (8.7) for the reachable target	$1.1463e^{+04}$	$4.8975e^{+03}$
System (8.7) for the unreachable target	$1.6545e^{+04}$	$9.9434e^{+03}$

Table 8.1: Performance indices for step changes in the target for system (8.7).

It can be clearly seen from Table 8.1 that the value of performance indices J with advance knowledge is lower than those without advance knowledge. This provides an insight into the usefulness of using advance knowledge in OMPC design.

8.3 Nominal oil gas plant feed-forward control design

In this section, we will implement the feed-forward design algorithm described in Subsection 8.1.3 in a subsystem model of the gas treatments process, as has been recently studied in [4]. This model is the second column of the process shown in Figure 8.8.

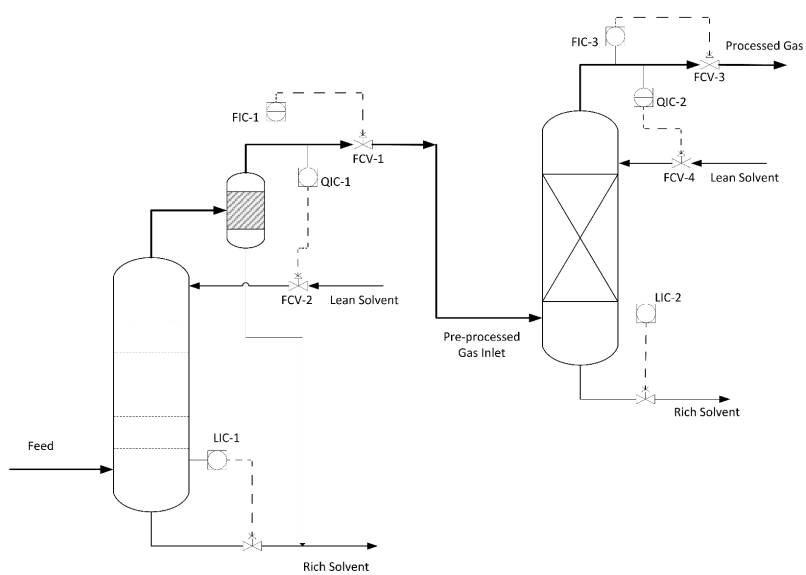

Figure 8.8: Two column gas treatment process [4]

Subsection 8.3.1 presents the model description of the gas treatment model. Subsection 8.3.2 demonstrates the robust design for reachable targets while Subsection 8.3.3 examines the robust design for unreachable targets.

8.3.1 Gas treatment model

The process consists of two columns; the first column is for gas treatment and the second column for gas purification. In this study the focus will be on the second column in which the raw gas feed, which is treated in the first column, is purified [4].

There are two outputs from the second column process which must be controlled: the first output is the throughput gas flow and the second output is the column outlet gas quality. There are two manipulated variables (control inputs): the column gas outlet flow through FCV-3 and the column solvent input flow through FCV-4. The discrete transfer function matrix of the model is given by:

$$G_c = \begin{bmatrix} \frac{-0.1905z^{-5}}{1-0.9683z^{-1}} & \frac{-0.1905z^{-6}}{1-0.9429z^{-1}} \\ \frac{-0.1905z^{-8}}{1-0.9848z^{-1}} & \frac{-0.1905z^{-6}}{1-0.926z^{-1}} \end{bmatrix} \tag{8.14}$$

The model can be described in state space form as a 29-state dimensional model with system matrices $A,\ B,\ C$ and D (see Appendix B.2).

The goal of the gas treatment is to maintain the gas flow as well as its quality at a specified level. This requires a priori information on the gas feed changes. In this section, we will deploy the proposed OMPC feed-forward algorithm, described in Section 8.1.2, in the corresponding column two model to maintain both the gas flow and its quality as desired.

A large set point change in the gas raw feed can directly affect the outlet gas law and its quality, particularly, when the steady-states are close to the maximum constraints. This rapid change in the gas raw feed may cause target infeasibility as discussed in Section 5.1. Therefore, we will show how easily the proposed algorithm can handle the transient infeasibility or unreachable target through the effective use of the advance knowledge with regard to target changes.

8.3.2 Assuming the target is infeasible in transients but reachable in steady-state

In this subsection, we will study the scenario in which the target is infeasible in transients but will asymptotically become feasible.

For specified limits on the outputs, we can apply limits on the inputs using the definition discussed in Subsection 8.1.4. To do this, we assume that both outputs, the gas flow and its quality are tracking the targets of $r_1 = 1.5$ and $r_2 = 0.5$, respectively. Thus, the limits on the steady-state inputs and outputs, u_{ss} and y_{ss}, for this model dynamic can be obtained by $x_{ss} = k_{xr} * r_k$ and $u_{ss} = k_{ur} * r_k$ as:

$$\begin{bmatrix} -0.5 \\ 0 \end{bmatrix} \leq \begin{bmatrix} u_{1ss} \\ u_{2ss} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0.19 \end{bmatrix}; \qquad \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} y_{1ss} \\ y_{2ss} \end{bmatrix} \leq \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} \tag{8.15}$$

This implies that the targets will be feasible at steady-state when the limits on both the outputs and the control inputs exceed their corresponding steady-state values in equation (8.15).

Now, it is straightforward to apply limits to the inputs, such that the target is reachable or unreachable as follows.

Let us assume that a step change in the gas raw feed $r_1 = 1.5$ and a step change in gas quality is $r_2 = 0.5$ are introduced and the input and output constraints are given by:

$$\begin{bmatrix} -3 \\ -3 \end{bmatrix} \leq \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \leq \begin{bmatrix} 3 \\ 3 \end{bmatrix}; \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \leq \begin{bmatrix} 1.5 \\ 0.5 \end{bmatrix} \tag{8.16}$$

With these constraints (8.16), both targets, r_1 and r_2 are reachable since all of the steady-state values lie within the constraint limits.

A closed-loop simulation is performed for the system (8.14) with and without advance knowledge about the reference changes with the tuning parameters; control horizon is $n_c = 5$, the weighting matrix is $Q = C^T C$ and the weight in the input command is $R = 0.1I$ subject to the applied constraints (8.16) with reachable targets of $r_1 = 1.5$ and $r_2 = 0.5$. The advance knowledge is set to $n_a = 8$, as discussed in the previous subsection.

Figure 8.9 shows the closed-loop responses of both outputs; the outlet gas flow and the gas quality for the system (8.14) with and without advance knowledge.

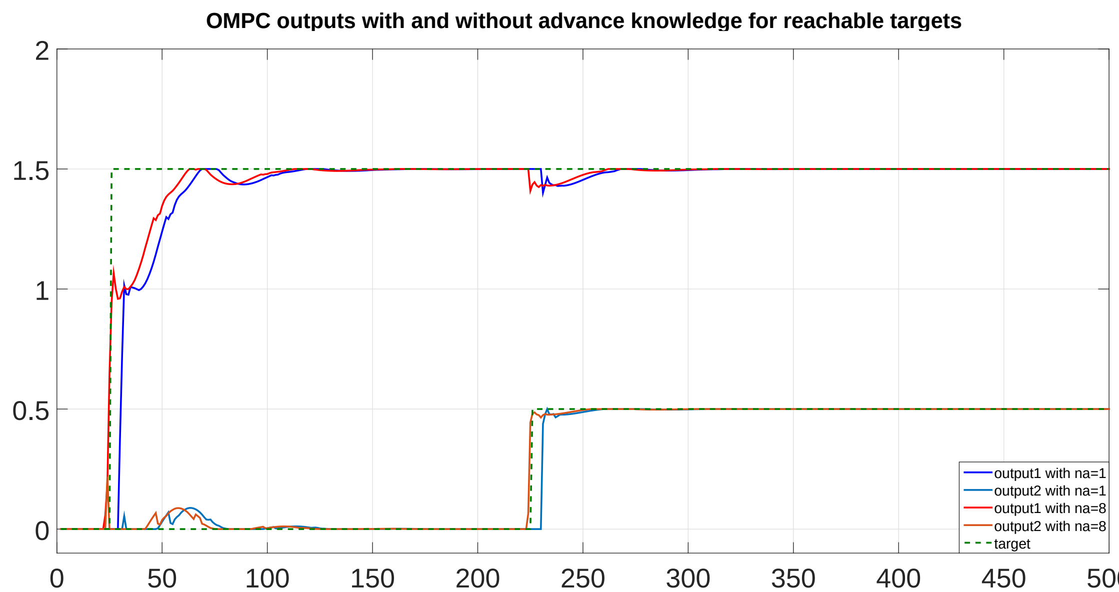

Figure 8.9: Closed-loop for the output step responses of system (8.14) for a reachable target with $n_a = 1$ and with $n_a = 8$

It is clear that the output responses with advance knowledge are faster and better than those without advance knowledge. This can be confirmed by computing the settling time in the sample time unit, for the corresponding responses as shown in Table 8.2. It can also be seen that the level of the interaction between the output variables, with advance knowledge, is slightly lower than those without advance knowledge.

	Settling time with $n_a = 1$	Settling time with $n_a = 8$
Output 1 for process (8.14)	90	80
Output 2 for process (8.14)	230	225

Table 8.2: Settling time of the closed-loop response for system (8.14) with and without advance knowledge.

It will be noted that the settling time for both output 1 and output 2 with $n_a = 8$ is lower than that for those with $n_a = 1$. This indicates the benefits of using advance knowledge about target changes. Further evidence of the advantages of using advance knowledge can be also illustrated in a tabular form, as shown in Table 8.3.

Figure 8.10 shows the evolution of the control inputs for the column two process for reachable targets with and without advance information about the reference changes.

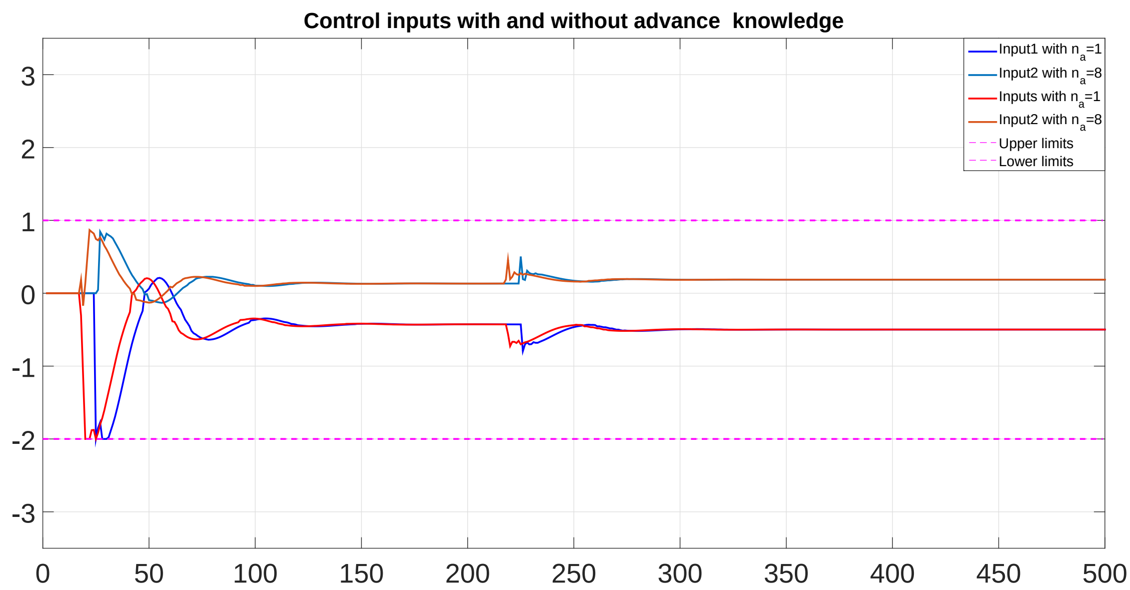

Figure 8.10: The evolution of the input commands of system (8.14) for a reachable target with $n_a = 1$ and with $n_a = 8$

It can be noticed that the constraints for the second input with and without advance knowledge do not offer any saturation while those for the first input do. This is because both their upper and lower limits exceed the corresponding steady-state values. Another observation is that the input constraints saturation with advance knowledge are recovered earlier than those without advance knowledge.

Figure 8.11 shows the evolution of the perturbations about the optimal at each sample with and without advance information about the reference changes.

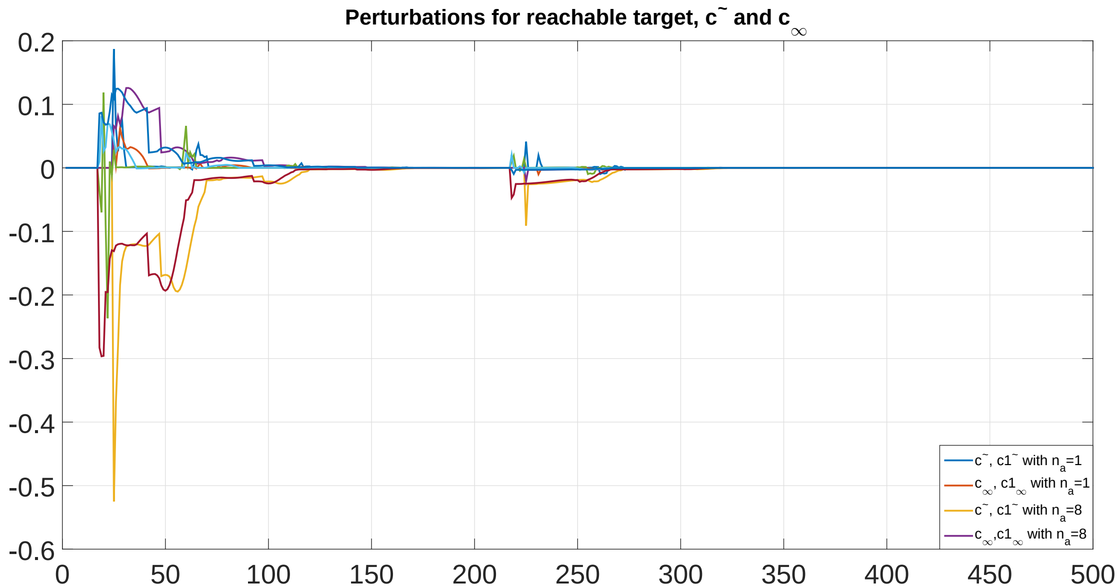

Figure 8.11: The evolution of the perturbations a bout optimal for a reachable target with $n_a = 1$ and with $n_a = 8$

It can be also seen that $\tilde{c}$ are non-zero only at times of set point changes due to constraint saturations. The key observation here is that perturbation levels with advance knowledge are lower than those without advance knowledge. It will be noted that the perturbations c_∞ are non-zero during transients but become zero for longer since both targets r_1 and r_2 are infeasible in transients but asymptotically become reachable.

The above discussion shows the advantages of using advance knowledge in the feed-forward design. Further evidence of the advantages of the effective use of advance knowledge in the feed-forward design can be shown through comparing the Performance indices, J with and without advance knowledge is shown in Table 8.3.

8.3.3 Assuming the target is unreachable

In this subsection, will study the scenario where the target is unreachable at steady-state. To do this, we assume that a step change in the gas raw feed of $r_1 = 1.6$ and a step change in the gas quality of $r_2 = 0.3$ are introduced. In this case, the target, r_1, is unreachable since its amplitude lies outside the output limits.

Next, we perform a closed-loop simulation for the system (8.14) with and without advance knowledge about the reference changes subject to the applied constraints (8.16), with tuning parameters of control horizon $n_c = 5$, a weighting matrix $Q = C^T C$ and the weight in the input command of $R = 0.1I$. The advance knowledge is set at $n_a = 8$, as discussed in the previous subsection.

Figure 8.12 shows the closed-loop responses of both outputs: the outlet gas flow and the gas quality for the system (8.14) with and without advance knowledge.

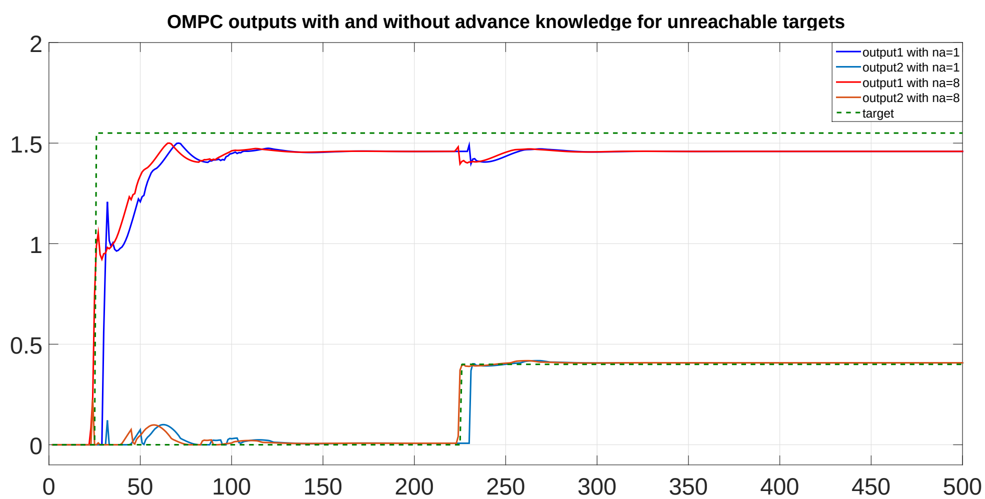

Figure 8.12: Closed-loop for the output step responses of system (8.14) for reachable target with $n_a = 1$ and with $n_a = 8$

It is clear that, the first target, $r_1 = 1.6$, is unreachable while the second target, $r_2 = 0.3$, is reachable. This is because, with r_1, the corresponding output steady-state value lies outside the output limits, whereas, the steady-state output with r_2 lies within them. However, the first output can track an artificial target which is as close as possible to the true target while the second output can smoothly track the true corresponding target. It can also be seen that the output responses with advance knowledge are faster and better than those without advance knowledge. This can be ensured by computing the rise time in the sample time for the corresponding responses, as shown in Table 8.2.

Figure 8.13 shows the evolution of the input commands for the column two process for reachable target with and without advance information about the reference changes.

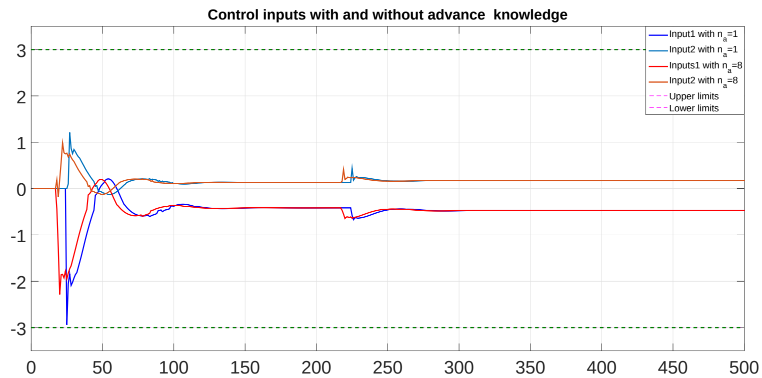

Figure 8.13: the evolution of the input commands of system (8.14) for a reachable target with $n_a = 1$ and with $n_a = 8$

It is clear that the control inputs with advance knowledge are lower than those with advance knowledge. The key observation here is that, although the targets are unreachable, the constraints for both inputs are not active.

Figure 8.14 shows the evolution of the perturbations about the optimal at each sample for system (8.14) with and without advance information about the reference changes.

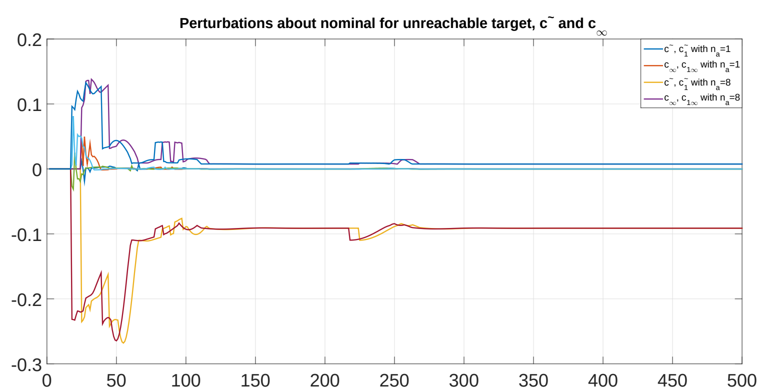

Figure 8.14: the evolution of the perturbations about the optimal for a reachable target with $n_a = 1$ and with $n_a = 8$

It can be seen that the perturbations c_∞ are non-zero at the time of the step changes and at steady-state since both targets are unreachable in both transients and steady-states. The key point here is that the level of the perturbations with advance knowledge is slightly lower than that with no advance knowledge. It can be also noticed that the perturbations $\tilde{c}$ are non-zero at the time of the target changes since the input constraints are active at that time.

The above discussion shows the advantages of using advance knowledge in feed-forward design. Further evidence of the advantages of the effective use of advance knowledge in the feed-forward design can be shown through the performance indices for the gas process with and without advance knowledge, for both reachable and unreachable targets, as shown in Table 8.3.

	J with $n_a = 1$	J with $n_a = 8$
System (8.14) for reachable targets	22.7943	9.8937
System (8.14) for unreachable targets	27.729	15.1

Table 8.3: Performance indices for step changes in the target for system (8.14)

It is obvious that the performance indices with advance knowledge ($n_a = 8$) are lower than those without advance knowledge ($n_a = 1$). This indicates that the systematic use of advance knowledge in feed-forward design is beneficial.

8.4 Robust feed-forward design: Parametric uncertainty

In this section, we will study the implementation of the proposed feed-forward design of Algorithm 6.3, described in Subsection 8.1.3, regarding a benchmark problem with a parametric uncertainty. Subsection 8.4.1 presents the model description of the uncertain benchmark problem. Subsection 8.4.2 demonstrates the robust design for reachable targets while Subsection 8.4.3 examines the robust design for unreachable targets. Subsection 8.4.4 demonstrates how effectively the proposed algorithm can handle model uncertainty.

8.4.1 Process with model uncertainty

We consider a process that contains a modelling uncertainty. This process is a two-mass-spring system, adopted from [168]. The process is described in Figure 8.15.

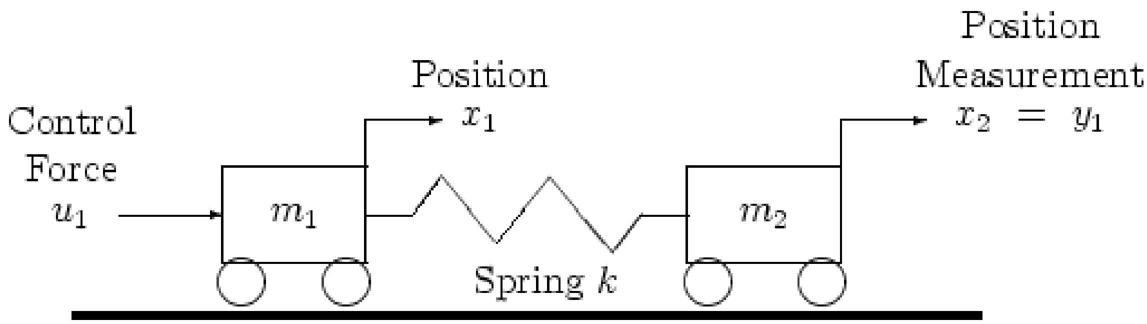

Figure 8.15: The uncertain mass-spring-system [168]

The system consists of two masses, the mass of body 1 and the mass of body 2, m_1 and m_2, respectively, and a spring with a stiffness k. The discrete state space representation of the process model can be described in terms of the masses and the spring constant (stiffness) as follows.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ v_1(k+1) \\ v_2(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ -0.1k/m_1 & 0.1k/m_1 & 1 & 0 \\ 0.1k/m_2 & -0.1k/m_2 & 0 & 1 \end{bmatrix}}_{Ap} \begin{bmatrix} x_1(k) \\ x_2(k) \\ v_1(k) \\ v_2(k) \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0.1/m_1 \\ 0 \end{bmatrix}}_{Bp} u_1(k) \quad (8.17)$$

$$y_1(k) = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}}_{Cp} \begin{bmatrix} x_1(k) \\ x_2(k) \\ v_1(k) \\ v_2(k) \end{bmatrix} \quad (8.18)$$

where $x_1(k)$ and $x_2(k)$ are the positions of body 1 and body 2, respectively, and $v_1(k)$ and $v_2(k)$ are the velocities of body 1 and body 2, respectively.

The variables Ap, Bp and Cp are the system state space parameters. The manipulated variable is the control force, u_1 and the output that must be controlled, $y_1(k)$, is the position measurement, $x_2(k)$.

Assuming that the only parametric uncertainty is the spring constant(stiffness), k and it varies between $0.5 \le k \le 2$ such that the system is stable and $m_1 = m_2 = 1$ [168], thus, the poly-topic uncertainty model of the system can be described as:

$$A = Co\left\{\underbrace{\begin{bmatrix} 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \\ -0.050.05 & 1 & 0 & \\ 0.05 & -0.05 & 0 & 1 \end{bmatrix}}_{A_1}, \underbrace{\begin{bmatrix} 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \\ -0.2 & 0.2 & 1 & 0 \\ 0.2 & -0.2 & 0 & 1 \end{bmatrix}}_{A_2}\right\} \tag{8.19}$$

.

$$B = B_1 = B_2 = \begin{bmatrix} 0 \\ 0 \\ 0.1 \\ 0 \end{bmatrix}, \tag{8.20}$$

.

A nominal model is assumed to be $A = 0.6A_1 + 0.4A_2$ and $B = B_1 = B_2$ is used to define the feedback controller, $K = \begin{bmatrix} 1.1516 & -0.2322 & 1.6733 & 0.6907 \end{bmatrix}$, as the LQ-Optimal for $Q = C^T C$ and $R = I$.

The goal of the design is to maintain body 2 in a predefined position in order to obtain the desired performance, in spite of the variation in the spring stiffness (model uncertainty).

8.4.2 Robust feed-forward design assuming a reachable target

In this subsection, we will discuss the OMPC tracking for reachable target. In order to demonstrate the robust tracking performance for the corresponding system for a reachable target, we assume that the position of body 2 must track a reference (target) of $r_k = 2$; thus, for the corresponding model dynamics, the steady-state values for x_{ss} can be computed using $x_{ss} = k_{xr} * r_k$, as follows.

$$k_{xr} = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}^T \implies x_{ss} = k_{xr} * r_k = \begin{bmatrix} 2 & 2 & 0 & 0 \end{bmatrix}^T * r_k \tag{8.21}$$

Thus, we can apply limits to the inputs and states for the corresponding model dynamics, such that the target is reachable, as discussed in Subsection 8.1.4. These limits can be given by:

$$-1 \leq u \leq 1; \quad \begin{bmatrix} -5 & -5 & -20 & -5 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 5 & 5 & 5 & 5 \end{bmatrix}^T \tag{8.22}$$

Now, we will perform a closed-loop simulation for the system of (8.17) with and without advance knowledge about target changes, assuming that the target is feasible. In this simulation, we assume that the real process is defined through the system parameters of $k = m_1 = m_2 = 1$. It will be noted that the real process is not the same as that for the nominal model ($Ap \neq A$), so the simulation is performed for robustness against the model uncertainty.

In order to consider the advance knowledge, we assume that the position of body 2 must track a reference (target) of $r_k = 2$, but with a priori information about future target changes. Therefore, we need to choose appropriate advance knowledge, n_a, as follows.

Since the corresponding system of (8.17) has a slow dynamics property, as shown in Figure 8.16, the appropriate advance knowledge can be obtained by using Algorithm 4.1 to be very large. However, we can argue that, as discussed in Chapter 4, for the constrained case, one can choose a lower n_a , which is close to n_c, to ease constraint handling. In this case, $n_a = 8$ can be useful. This can be demonstrated through a closed-loop simulation for the system of (8.17) with advance knowledge of $n_a = 8$ and the tuning parameters can be set as follows. The control horizon is $n_c = 5$, the weighting matrix on the states is $Q = C^T C$ and the weight in the input command is $R = I$.

Figure 8.16 shows the output step responses of the uncertain system (8.17) for a reachable target, with $n_a = 1$ and $n_a = 8$.

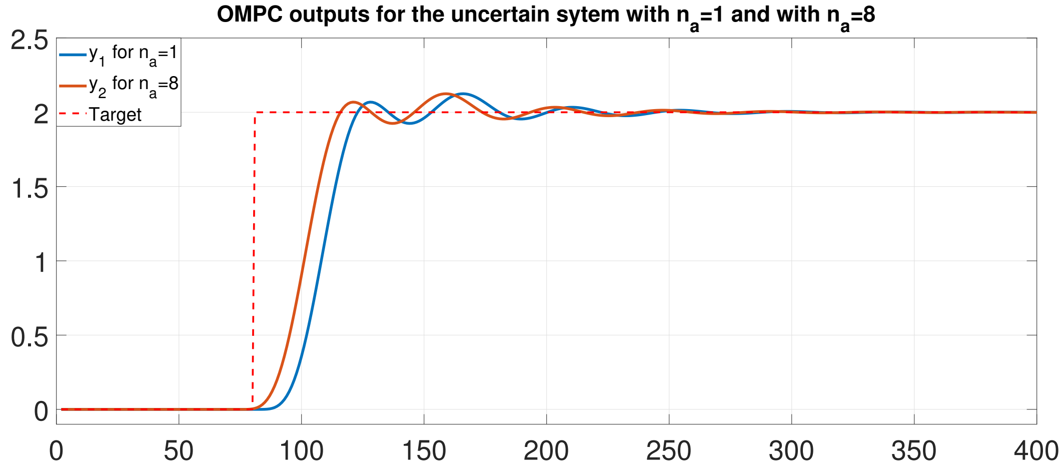

Figure 8.16: Closed-loop output step responses of the uncertain system (8.17) with advance knowledge of $n_a = 1$ and $n_a = 8$

It is clear that, despite the presence of model uncertainty, the output responses of the position of body 2 can asymptotically track the reference input (target), $r_k = 2$, with a slight oscillation. The key observation here is that the response with advance knowledge is faster than that without advance knowledge.

The control input evolutions (upper plot) and the input perturbations about th optimal (lower plot) for the uncertain system are also shown in Figure 8.17.

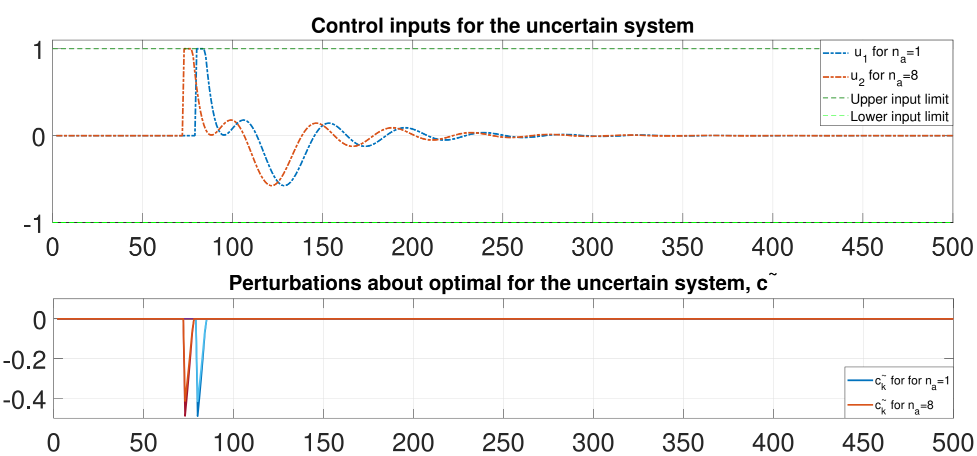

Figure 8.17: The evolution of the inputs and perturbations of the uncertain system (8.17) with advance knowledge $n_a = 1$ and $n_a = 8$.

It can also be seen that the applied forces (control inputs) are active at the time of the step change and then move to their corresponding steady-state. It will be noted that the input perturbations, $\tilde{c}_k$ are non-zero at the transients and become zero for a longer period of time. This indicates that there exist deviations from the unconstrained optimal only in the transients. It is also noticeable that the saturation recovery for the input with advance knowledge occurs earlier than is the case without advance knowledge. This discussion provides insights into how smoothly the proposed algorithm can handle the constraints for reachable targets and effectively use information about future target changes for the uncertain case.

8.4.3 Robust feed-forward design assuming unreachable targets

In this subsection, we will demonstrate the tracking OMPC for unreachable targets. To achieve this, we will consider the input and state constraints as:

$$-1 \leq u \leq 1; \quad \begin{bmatrix} -5 & -5 & -8 & -5 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 1.5 & 1.5 & 5 & 5 \end{bmatrix}^T \tag{8.23}$$

and assume that the position of body 2 must track a reference (target) of $r_k = 1.6$. Using this target, the corresponding steady-state and input can be computed with $x_{ss} = k_{xr} * r_k$ and $u_{ss} = k_{ur} * r_k$ as:

$$x_{ss} = k_{xr} * r_k = \begin{bmatrix} 1.5 & 1.5 & 0 & 0 \end{bmatrix}^T \tag{8.24}$$

Since the applied target lies outside the applied steady-state output (8.24), therefore, it cannot be reached (unreachable).

A closed-loop simulation is performed for the system of (8.17) for unreachable targets with and without advance knowledge about the reference changes. In this simulation, we assume again that the real process is defined by the system parameters of $k = m_1 = m_2 = 1$. It will be noted that the real process is not the same as in the nominal model ($Ap \neq A$), so the simulation is performed for robustness against the model uncertainty.

The tuning parameters can be set as follows. The control horizon is $n_c = 5$, the weighting matrix $Q = C^T C$ and the weight in the input command $R = I$. The advance knowledge is taken to be $n_a = 10$ which can also be useful for this system as discussed in reference to the previous simulation.

Figure 8.18 shows the output step responses for the uncertain system (8.17) for an unreachable target with $n_a = 1$ and $n_a = 10$.

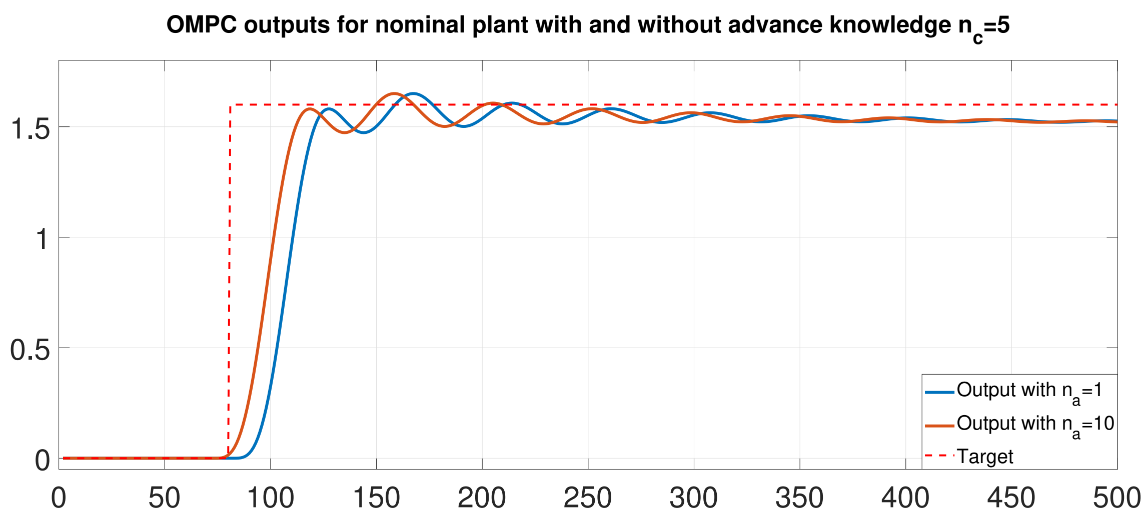

Figure 8.18: Closed-loop for output step responses of the uncertain system (8.17) with $n_a = 1$ and $n_a = 10$

Once again, although the system is uncertain, it is clear that the target $r_k = 1.6$, cannot be reached as expected but that the position of body 2 tracks an artificial target $\hat{r}_k$, in this case $\hat{r}_k = 1.5$, which is close as soon as possible to the true target ($r_k = 1.6$). The key observation here is that the response with advance knowledge is better than that without advance information about the future target.

The evolution of the inputs (upper plot) and the perturbations about the optimal (lower plot) for the uncertain system are also shown in Figure 8.19.

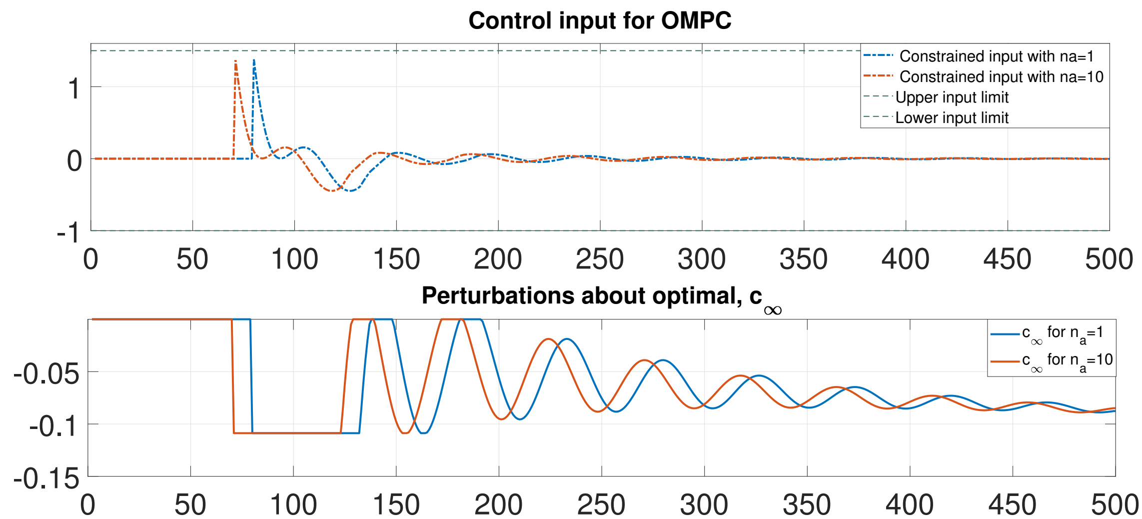

Figure 8.19: The inputs and perturbations of the uncertain system (8.17) with $n_a = 1$ and $n_a = 10$

It can also be seen that the control inputs (upper plot) that are used for the system with and without advance knowledge, are inactive at the time of the step change and then moves to its corresponding steady-state. It will be noted that the term c_∞ for both cases (lower plot) is non-zero for the long term. This indicates that there exist deviations from the actual unreachable target. This provides an insight in that the proposed algorithm can smoothly handle unreachable targets in the presence of model parameter uncertainty.

Evidence for the benefits of using advance knowledge for robust tracking is shown in Table 8.4 by performing a comparison between the performance indices, J of the uncertain system with, $n_a = 1$ and those with $n_a = 8$ for reachable and unreachable targets.

	J with $n_a = 1$	J with $n_a = 8$
System (8.17) for reachable target	103.9	80.07
System (8.17) for unreachable target	427.5	373.44

Table 8.4: Performance indices for step changes in the target of the uncertain system (8.17) for reachable and unreachable targets.

It can be seen from Table 8.4 that the performance indices values of the corresponding system with advance knowledge are lower than those for that with no advance knowledge. This indicates that the appropriate use of advance knowledge on the feed-forward design is beneficial.

8.4.4 Robust feed-forward design for different uncertain parameters

In this subsection, we will show how effectively can the proposed robust OMPC algorithm handle model uncertainty with the presence of input and output constraints and the inclusion of advance knowledge about future target changes. We will perform a closed-loop simulation of the step response of the corresponding process (8.19) for three different selected uncertain (spring constant, k) parameters. This advance knowledge is taken to be $n_a = 8$ and the target is $r_k = 2$.

The state space parameters of the real process are assumed to be as follows: $A_p = A_1$ ($k = 0.5$), $A_p = 0.5A_1 + 0.5A_2$ ($k = 1.25$) and $A_p = A_2$ ($k = 2$). The tuning parameters are set as: $n_c = 5$, $R = I$ and $Q = C^T C$. The input and state constraints are.

$$-1 \leq u \leq 1; \qquad \begin{bmatrix} -5 & -5 & -10 & -5 \end{bmatrix}^T \leq x \leq \begin{bmatrix} 5 & 5 & 5 & 5 \end{bmatrix}^T \tag{8.25}$$

Figure 8.20 shows the output responses of the uncertain system (8.17), y_1, y_2 and y_3 for three selected uncertain parameters of $k = 0.5,\ 1.25$ and 2, respectively, with advance knowledge about future target changes of $n_a = 8$, assuming that the target is reachable in both the transient and steady-states.

It is clear that the position of body 2, x_2, follows the applied target of $r_k = 2$ for the three selected spring constant parameters asymptotically with good performance. This indicates that the proposed algorithm can effectively handle the model uncertainty as well as the advance knowledge for the tracking scenario.

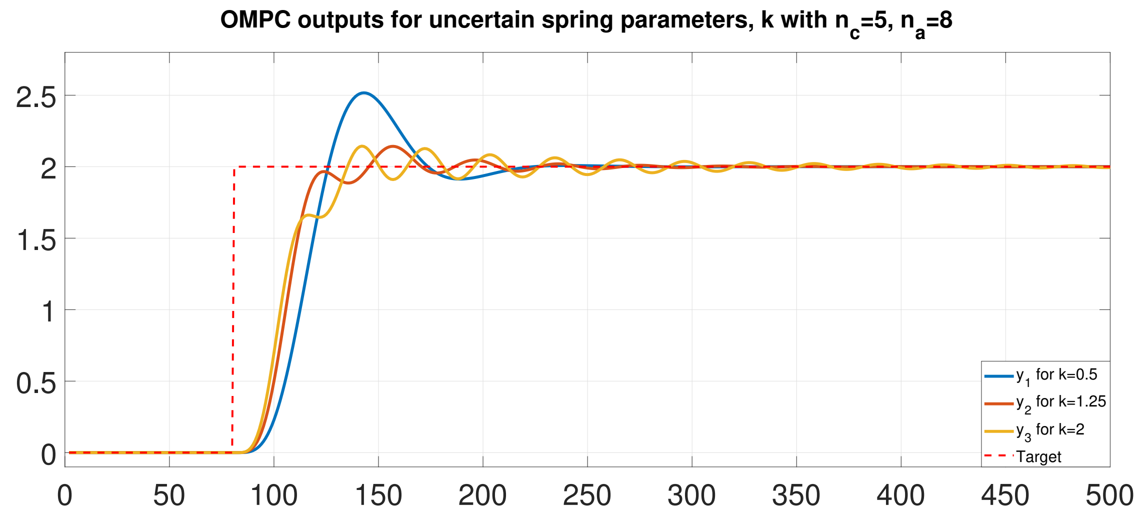

Figure 8.20: Closed-loop for output step responses of the uncertain system (8.17) with $n_a = 8$

Figure 8.21 shows the OMPC control inputs (forces) applied to the mass of body 1, m_1 of the corresponding system, u_1, u_2 and u_3 for three uncertain spring constant parameters of $k = 0.5, 1.25$ and 2, respectively,

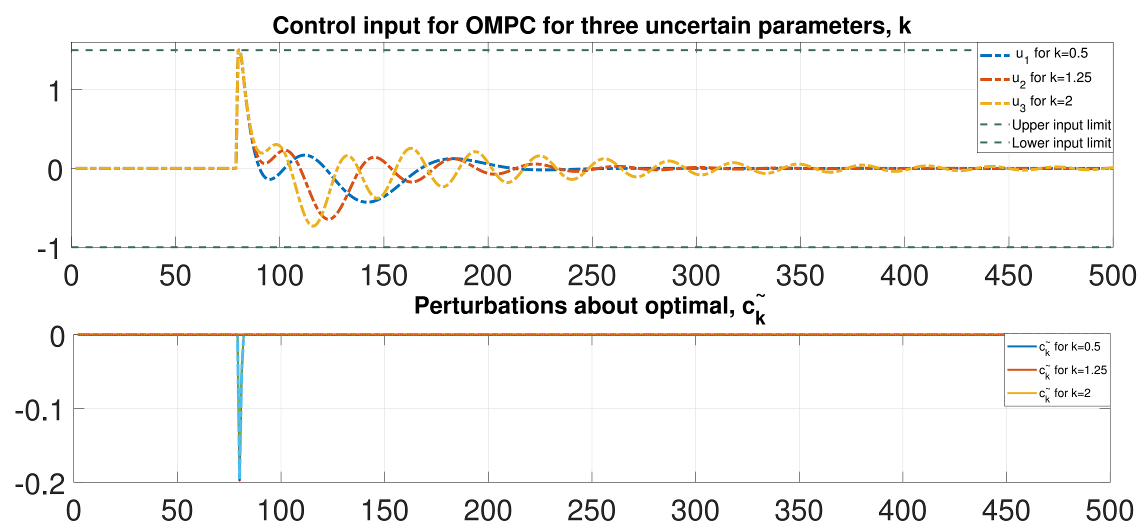

Figure 8.21: The evolution inputs and perturbations of the uncertain system (8.17) with $n_a = 8$

It can be observed that the input constraints for all of the spring constant uncertain parameters are active at the transients. This makes the input perturbations, c_k non-zero at that time. It is also observed that the input perturbation are zero for a longer period of time since the control input constraints are inactive at that time, as shown in the plot.

8.5 Discussion and conclusions

This chapter studied the applicability to industrial processes of the proposed OMPC algorithms, for feed-forward design with the use of future information about target changes. The proposed algorithms were examined in reference to three different industrial processes.

It has been shown that the proposed OMPC algorithm can be easily implemented in various industrial processes, from small SISO to large scale MIMO processes.

Regarding feasibility and stability, the obtained results show the effectiveness of the proposed Algorithm 5.1. The results also show the ability of the proposed control structure to handle process constraints as well as advance knowledge for both the reachable and unreachable targets. The key point here is that the controller tackles the transient target infeasibility by artificially adding an additional d.o.f, c_∞ rather than increasing the control horizon, n_c

Regarding the robust case, the results show the efficacy of the proposed Algorithm 6.3 in dealing with constrained uncertain systems with an effective use of advance knowledge about the target changes for both reachable and unreachable targets.

Regarding the use of advance knowledge in the proposed feed-forward design, both Algorithms 5.1 and 6.3 are simple and easy to implement in various industrial processes, thereby providing better tracking performance compared to that without using future information about target changes. The benefits of using advance knowledge were demonstrated through comparing the output responses with advance knowledge and those without advance knowledge.

It has been shown that, through the use of advance knowledge, the output responses are faster than those without advance knowledge. The advantage of using advance knowledge is also demonstrated by means of the input control evolution plot, in which less input effort is used when the advance knowledge is included. Further evidence of the benefits of using advance knowledge within tracking OMPC algorithms was shown in tabular form, where the performance indices of the corresponding systems with advance knowledge were demonstrated to be lower than those without advance knowledge.

Chapter 9

CONCLUSIONS AND FUTURE WORK

The thesis discusses how to improve tracking with predictive control, using dual-mode or OMPC approaches. In this chapter, we will present the conclusions of the thesis followed by future work recommendations. Section 9.1 presents a summary of the contribution of this thesis. Section 9.2 presents an overall summary of the conclusions, while Section 9.3 identifies, the weaknesses found in the approaches proposed in this thesis and proposes future work.

9.1 Thesis contribution

Following a brief review of the literature on the approaches to tracking within MPC, it is clear that very few researchers have utilised advance information on target changes, and the common assumption is that no advance information is available. The thesis makes several important contributions, which relate to MPC tracking improvement, particularly, the use of advance information about target changes in a feed-forward design, using both QP and mp-QP solutions. These contributions are listed as follows:

- It has been shown that the default feed-forward arising from a conventional MPC algorithm may be deficient because the assumptions implicit in the optimisation are relatively limited and only valid for a fixed target. If the future target is changed, then the optimisation and degrees of freedom within it require essential modification.
- The thesis defined an algorithm that can make effective use of advance knowledge of target changes, by embedding this into the on-line optimisation and evaluating the

approach using Matlab simulation. Another consideration is that the future values of the target can be treated as states in a parametric optimisation with an associated reduction in computational complexity.

- In practice, there is a scenario in which the constraints are active in the steady-state, which can prevent the achievement of the desired target. Recent work has proposed an artificial target which is reachable, but the challenge here is how to compute and choose this target, as well as how to incorporate it into the MPC optimisation. Therefore, the thesis introduces a novel algorithm that computes and chooses an artificial target and incorporates it into the performance index using dual-mode or OMPC algorithms to demonstrate the impact on both terminal constraint and feasibility.

- It has been shown that uncertainty can cause a loss of feasibility, which would result in the MPC becoming undefined. Therefore, the thesis analysed the existing approaches that are robust to uncertainty and proposed several modifications to guarantee this robustness. Recent work has been carried out in this area, such as a robust invariant set approach, but this is limited to specific scenarios. The thesis defined an alternative set (Robust MCAS), which can effectively handle both reachable and unreachable targets. The proposed MCAS approach, is flexible and makes it easy to deploy the OMPC algorithms.

- It is clear that including advance information about targets increases the dimension of the parameter space for an mP-QP approach to predictive control. It is recognised that mP-QP is often impractical for use with large parameter spaces and thus one may infer that mP-QP will tend to be ineffective for use in conjunction with advance knowledge scenarios. Nevertheless, this thesis introduced several reformulations of a typical dual-mode (OMPC) algorithm which can, to a limited extent, overcome problems related to dimension growth.

- Finally, the thesis outlined the consequences of including core parameters, such as the target and the current input, within the complexity of any parametric solutions. Moreover, other important issues that have been demonstrated include the influence of the allowance of the steady-state offset on the feasibility and computational complexity of parametric solutions.

9.2 Overall conclusions

The thesis investigated the impact of advance knowledge on system performance using OMPC algorithms. It has shown that this response, where there exists too much advance knowledge, may result in earlier anticipation than desired, whereas too little may lead to a delayed response. A critical insight is that it is impossible to provide generic, useful and theoretical results regarding the optimal amount of advance knowledge to use, due to the dependence on this amount of tuning, system dynamics, and constraints. At this point, the thesis introduces an algorithm which makes it possible to choose sensible value for the advance knowledge for any given system. The algorithm provides simple guidance to evaluate this amount, that appears to be effective in the unconstrained case and would be easy to deploy in general.

In order to counteract the constraints, it is better to use an amount of advance knowledge close to the number of d.o.f. At this point, the thesis introduces an OMPC algorithm, which handles constraints alongside advance knowledge. This algorithm has embedded the default unconstrained feed-forward in the optimization with a small value of advance knowledge. The algorithm provides good results with optimal behavior and easy constraint handling, thus in a manner that is useful to a field engineer who simply wishes to ensure they are approximately correct. The efficacy and simplicity of this approach are demonstrated using $MATLAB$ 2017a simulation.

One important component required to deal with the constraints is the admissible set algorithm, so the thesis has developed an alternative to this algorithm, denoted as (MCAS) algorithm. It is shown that this algorithm can handle time variation in both reachable and unreachable targets. It is also shown, that it can be extended to be implemented effectively in uncertain systems to counteract both regulation and tracking problems. Moreover, the proposed algorithm may easily be used to solve parametric optimisation problems. The differences lie solely in the autonomous model structure and hence in the sample constraints.

At times, the desired target will be unreachable, in which scenarios a default MPC algorithm becomes ill-defined. This thesis proposes a simple alternative, which caters for both transient and permanent infeasibility in the target without requiring changes to the algorithm on-line. Moreover, it shows how, even in this case, the systematic embedding of advance information is straightforward and beneficial.

The thesis shows how the previous contributions can be extended in a straightforward manner to cater for parameter uncertainty by proposing algorithms for uncertain (LPV) systems, for both reachable and unreachable target scenarios. The algorithms provide robust guarantees of feasibility and convergence while utilising a simple QP optimisation on-line.

The thesis introduces an algorithm, which demonstrates how to reduce the dimension of parametric space for parametric approaches. In general, the study of reducing target information can simplify the parametric solution, but here it is shown that, for most systems, the result is as expected, but is not true for all systems. Moreover, the weight R may influence the number of regions and indeed many other factors, such as the model dynamics. Therefore, we can argue that analytic results are impossible and that we cannot apply this concept to a specific system. In general, this best expectation is not guaranteed.

The thesis investigates the impact of including the target and the steady-state offset (artificial perturbation) into the parametric solution to the feasibility issue in MPC. It has shown that no obvious pattern exists, but of course one could argue that including c_∞ gives the much larger feasible volume in general for the same number of d.o.f, so, for equivalent volumes of feasible region, it is likely that using c_∞ will result in far fewer parametric regions.

In summary, the thesis shows that tracking within the predictive control is improved through the design of an efficient feed-forward compensator using OMPC algorithms with advance knowledge information of the target changes effectively. For parametric approaches, however, there may not exist an effective way to utilise advance information about target changes.

9.3 Future work and weaknesses

9.3.1 Weaknesses

The main weaknesses identified in this research are listed as follows.

- The admissible set (MCAS) constructed in this thesis, may be a challenge because of the dependence on the processors. Moreover, in the case of constraints on the boundary, the MCAS may not converge in a finite limit.
- For large dimensional systems, the parametric solutions discussed in Chapter 7 may be too complicated to compute.

9.3.2 Future work recommendations

Future work might consider the following proposals:

- The use of the steady-state offset for improving feasibility is discussed in Chapter 7.5. This discussion focuses on proposals which improve performance and feasibility but, in further study, it is recommended to choose a more systematic performance index and allow alternatives to the steady-state offset in the optimisation problems.
- The proposed algorithms proposed in Chapters 4-6, are difficult to deploy with the decentralised predictive control system since the interaction between the controllers must be precise. Accordingly, we must consider how to modify these and apply them to the decentralised predictive control system.
- The work done in this thesis could be applied on hardware tools such as PLC to prove the efficacy of the proposed algorithms.
- Finally, the repercussions of the parametric solutions outlined in Chapter 7 might be expanded by undertaking a more comprehensive and wide-ranging set of case studies.

BIBLIOGRAPHY

[1] R. J. M. Afonso and R. K. H. Galvão. Infeasibility handling in constrained MPC. In *Frontiers of Model Predictive Control*. InTech, 2012.

[2] S. Aghaee, Y. Zakeri, and F. Sheikholeslam. Offset-free control of constrained linear systems using model predictive control. In *Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on*, pages 973–979, 2008.

[3] S. Aghaei, F. Sheikholeslam, M. Farina, and R. Scattolini. An MPC-based reference governor approach for offset-free control of constrained linear systems. *International Journal of Control*, 86(9):1534–1539, 2013.

[4] Y. Al-Naumani. *MPC for Upstream Oil & Gas Fields a practical view*. PhD thesis, University of Sheffield, 2017.

[5] I. Alvarado, D. Limon, T. Alamo, and E. F. Camacho. Output feedback Robust tube based MPC for tracking of piece-wise constant references. In *2007 46th IEEE Conference on Decision and Control*, pages 2175–2180, 2007.

[6] I. Alvarado, D. Limon, A. Ferramosca, T. Alamo, and E. F. Camacho. Robust tubed-based MPC for tracking applied to the quadruple-tank process. In *2008 IEEE International Conference on Control Applications*, pages 305–310, 2008.

[7] P. J. Antsaklis and A. N. Michel. *A linear systems primer*, volume 1. Birkhäuser Boston, 2007.

[8] M. Balandat. Constrained robust optimal trajectory tracking: Model predictive control approaches. *Master's thesis, Technische Universitat Darmstadt*, 2010.

[9] M. Baotić. An efficient algorithm for multi-parametric quadratic programming. Technical report, ETH Zürich, Institut für Automatik, 2002.

[10] A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *Automatic Control, IEEE Transactions on*, 42(3):340–349, 1997.

[11] A. Bemporad and C. Filippi. Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. *Journal of optimization theory and applications*, 117(1):9–38, 2003.

[12] A. Bemporad and C. Filippi. An algorithm for approximate multiparametric convex programming. *Computational optimization and applications*, 35(1):87–108, 2006.

[13] A. Bemporad and M. Morari. Robust model predictive control: A survey. In *Robustness in identification and control*, pages 207–226. Springer, 1999.

[14] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit solution of model predictive control via multiparametric quadratic programming. In *American Control Conference, 2000. Proceedings of the 2000*, volume 2, pages 872–876, 2000.

[15] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[16] G. Betti, M. Farina, and R. Scattolini. An MPC algorithm for offset-free tracking of constant reference signals. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5182–5187, 2012.

[17] G. Betti, M. Farina, and R. Scattolini. A robust MPC algorithm for offset-free tracking of constant reference signals. *IEEE Transactions on Automatic Control*, 58(9):2394–2400, 2013.

[18] H. Bouhenchir, M. Cabassud, and M. L. Lann. Predictive functional control for the temperature control of a chemical batch reactor. *Computers & Chemical Engineering*, 30(6):1141 – 1154, 2006.

[19] E. F. Camacho and A. C. Bordons. *Model predictive control*. Springer Science & Business Media, 2013.

[20] M. Canale, L. Fagiano, and M. Milanese. Set membership approximation theory for fast implementation of model predictive control laws. *Automatica*, 45(1):45–54, 2009.

[21] D. Carrasco and G. Goodwin. Preview and feedforward in model predictive control: A preliminary robustness analysis. In *18th IFAC World Congress*, pages 185–190, 2011.

[22] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[23] X. Cheng and D. Jia. Robust stability constrained model predictive control. In *ACC*, volume 2, pages 1580–1585, 2004.

[24] L. Chisci, J. Rossiter, and G. Zappa. Systems with persistent disturbances: Predictive control with restricted. *Automatica*, 37(7):1019–1028, 2001.

[25] L. Chisci and G. Zappa. Dual mode predictive tracking of piecewise constant references for constrained linear systems. *International Journal of Control*, 76(1):61–72, 2003.

[26] D. Clarke. Generalized predictive control: A robust self-tuning algorithm. In *American Control Conference, 1987*, pages 990–995, 1987.

[27] D. Clarke. Application of generalized predictive control. *IFAC Proceedings Volumes*, 21(7):1–8, 1988.

[28] D. Clarke and R. Scattolini. Constrained receding-horizon predictive control. In *IEE Proceedings D (Control Theory and Applications)*, volume 138, pages 347–354, 1991.

[29] D. W. Clarke. *Advances in model-based predictive control*. Oxford ; New York : Oxford University Press, 1994.

[30] D. W. Clarke and C. Mohtadi. Properties of generalized predictive control. *Automatica*, 25(6):859–875, 1989.

[31] J. Compas, P. Decarreau, G. Lanquetin, J. Estival, N. Fulget, R. Martin, and J. Richalet. Industrial applications of predictive functional control to rolling mill, fast robot, river dam. In *Proceedings of IEEE International Conference on Control and Applications*, pages 1643–1655 vol.3, 1994.

[32] P. Cortés, G. Ortiz, J. I. Yuz, J. Rodríguez, S. Vazquez, and L. G. Franquelo. Model predictive control of an inverter with output *LC* filter for UPS applications. *IEEE Transactions on Industrial Electronics*, 56(6):1875–1883, 2009.

[33] C. R. Cutler and B. L. Ramaker. Dynamic matrix control - a computer control algorithm. In *joint automatic control conference*, number 17, 1980.

[34] F. A. Cuzzola, J. C. Geromel, and M. Morari. An improved approach for constrained robust model predictive control. *Automatica*, 38(7):1183–1189, 2002.

[35] N. de Oliveira and L. T. Biegler. Constraint handling and stability properties of model-predictive control. *AIChE journal*, 40(7):1138–1155, 1994.

[36] S. Dughman and J. Rossiter. Systematic and simple guidance for feed forward design in model predictive control. In *International Conference on Control Science and Systems Engineering*, volume 18, pages 1229–1234, 2016.

[37] S. Dughman and J. A. Rossiter. Efficient feed forward design within MPC. In *Control Conference (ECC), 2016 European*, pages 1341–1346, 2016.

[38] C. G. Economou, M. Morari, and B. O. Palsson. Internal model control: Extension to nonlinear system. *Industrial & Engineering Chemistry Process Design and Development*, 25(2):403–411, 1986.

[39] P. Falugi and D. Mayne. Model predictive control for tracking random references. In *Control Conference (ECC), 2013 European*, pages 518–523, 2013.

[40] A. Ferramosca, D. Limon, I. Alvarado, T. Alamo, and E. F. Camacho. MPC for tracking of constrained nonlinear systems. In *Proceedings of the 48h IEEE Conference on Decision and Control*, pages 7978–7983, 2009.

[41] A. Ferramosca, D. Limón, I. Alvarado, T. Alamo, F. Castaño, and E. F. Camacho. Optimal MPC for tracking of constrained linear systems. *International Journal of Systems Science*, 42(8):1265–1276, 2011.

[42] A. V. Fiacco and G. P. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*, volume 4. Siam, 1990.

[43] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, 2013.

[44] B. A. Francis and W. M. Wonham. The internal model principle of control theory. *Automatica*, 12(5):457–465, 1976.

[45] G. Frison and J. B. Jørgensen. Efficient solvers for soft-constrained MPC. In *18th Nordic Process Control Workshop*, 2015.

[46] S.-z. Gao, J.-s. Wang, and X.-w. Gao. Generalized predictive control of pvc polymerization based on piecewise affine. *Journal of chemical engineering of Japan*, 46(6):407–413, 2013.

[47] C. E. Garcia and M. Morari. Internal model control. a unifying review and some new results. *Industrial & Engineering Chemistry Process Design and Development*, 21(2):308–323, 1982.

[48] C. E. Garcia and A. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 46(1-3):73–87, 1986.

[49] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practice a survey. *Automatica*, 25(3):335–348, 1989.

[50] E. G. Gilbert, I. Kolmanovsky, and K. T. Tan. Nonlinear control of discrete-time linear systems with state and control constraints: A reference governor with global convergence properties. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 1, pages 144–149, 1994.

[51] E. G. Gilbert, I. Kolmanovsky, and K. T. Tan. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *International Journal of robust and nonlinear control*, 5(5):487–504, 1995.

[52] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic control*, 36(9):1008–1020, 1991.

[53] A. H. González, E. J. Adam, and J. L. Marchetti. Conditions for offset elimination in state space receding horizon controllers: A tutorial analysis. *Chemical Engineering and Processing: Process Intensification*, 47(12):2184–2194, 2008.

[54] G. Goodwin, D. Carrasco, D. Mayne, M. Salgado, and M. Seron. Preview and feedforward in model predictive control: Conceptual and design issues. In *18th IFAC World Congress*, pages 5555–5560, 2011.

[55] J. Gossner, B. Kouvaritakis, and J. A. Rossiter. Stable generalized predictive control with constraints and bounded disturbances. *Automatica*, 33(4):551–568, 1997.

[56] L. Grüne. NMPC without terminal constraints. *IFAC Proceedings Volumes*, 45(17):1–13, 2012.

[57] A. Gupta, S. Bhartiya, and P. Nataraj. A novel approach to multiparametric quadratic programming. *Automatica*, 47(9):2112–2117, 2011.

[58] R. Haber, U. Schmitz, and K. Zabet. Implementation of PFC (Predictive Functional Control) in a petrochemical plant. *IFAC Proceedings Volumes*, 47(3):5333–5338, 2014.

[59] W. W. Hager. Lipschitz continuity for constrained processes. *SIAM Journal on Control and Optimization*, 17(3):321–338, 1979.

[60] O. Härkegård. *Backstepping and control allocation with applications to flight control*. PhD thesis, Linköpings universitet, 2003.

[61] K. Holkar and L. Waghmare. An overview of model predictive control. *International Journal of Control and Automation*, 3(4):47–63, 2010.

[62] K. Horvath, E. Galvis, M. G. Valentin, and J. Rodellar. New offset-free method for model predictive control of open channels. *Control Engineering Practice*, 41:13–25, 2015.

[63] M. Hovd. Multi-level programming for designing penalty functions for MPC controllers. *IFAC Proceedings Volumes*, 44(1):6098–6103, 2011.

[64] J. C. Hung. Practical techniques for industrial control. In *Industrial Electronics, 1996. ISIE'96., Proceedings of the IEEE International Symposium on*, volume 1, pages 5–10, 1996.

[65] H. Jianbo, S. Manhong, and X. Jun. Predictive functional control and its application to missile control system. In *25 th International Congress of the Aeronautical Sciences*, 2006.

[66] T. A. Johansen, I. Petersen, and O. Slupphaug. On explicit suboptimal LQR with state and input constraints. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 1, pages 662–667. IEEE, 2000.

[67] C. N. Jones, M. Barić, and M. Morari. Multiparametric linear programming with applications to control. *European Journal of Control*, 13(2):152–170, 2007.

[68] E. A. Karl. A simple event-based PID controller. *IFAC Proceedings Volumes*, 32(2):8687 – 8692, 1999.

[69] E. Kerrigan and J. Maciejowski. Soft constraints and exact penalty functions in model predictive control. In *Proceedings of the UKACC International Conference on Control*, volume 2000, 2000.

[70] B. Khan and J. Rossiter. Generalised parameterisation for MPC. *Proceedings of the IASTED International Conference on Intelligent Systems and Control.*, 2011.

[71] M. V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[72] K. I. Kouramas, N. P. Faísca, C. Panos, and E. N. Pistikopoulos. Explicit/multi-parametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming. *Automatica*, 47(8):1638–1645, 2011.

[73] B. Kouvaritakis and M. Cannon. *Model predictive control*. Springer, 2016.

[74] B. Kouvaritakis, J. Rossiter, and A. Chang. Stable generalised predictive control: an algorithm with guaranteed stability. In *IEE Proceedings D (Control Theory and Applications)*, volume 139, pages 349–362. IET, 1992.

[75] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Trans. on Automatic Control*, 45(8):1545–1549, 2000.

[76] G. Kreisselmeier and R. Steinhauser. Systematic control design by optimizing a vector performance index. In *Computer Aided Design of Control Systems*, pages 113 – 117. Pergamon, 1980.

[77] H.-B. Kuntze, A. Jacubasch, J. Richalet, and C. Arber. On the predictive functional control of an elastic industrial robot. In *Decision and Control, 1986 25th IEEE Conference on*, pages 1877–1881, 1986.

[78] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. Multi-parametric toolbox (MPT). In *International Workshop on Hybrid Systems: Computation and Control*, pages 448–462. Springer, 2004.

[79] M. Kvasnica, J. Hledík, I. Rauová, and M. Fikar. Complexity reduction of explicit model predictive control via separation. *Automatica*, 49(6):1776–1781, 2013.

[80] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[81] Y. Lee, S. Park, M. Lee, and C. Brosilow. PID controller tuning for desired closed-loop responses for SISO systems. *Aiche journal*, 44(1):106–115, 1998.

[82] S. Li, H. Liu, and W. Fu. An improved predictive functional control method with application to PMSM systems. *International Journal of Electronics*, 104(1):126–142, 2017.

[83] J. S. Lim, J.-S. Kim, and Y. I. Lee. Robust tracking model predictive control for input-constrained uncertain linear time invariant systems. *International Journal of Control*, 87(1):120–130, 2014.

[84] D. Limon, T. Alamo, D. M. de la Pena, M. N. Zeilinger, C. Jones, and M. Pereira. MPC for tracking periodic reference signals. *IFAC Proceedings Volumes*, 45(17):490–495, 2012.

[85] D. Limon, I. Alvarado, T. Alamo, and E. Camacho. Robust tube-based MPC for tracking of constrained linear systems with additive disturbances. *Journal of Process Control*, 20(3):248–260, 2010.

[86] D. Limón, I. Alvarado, T. Alamo, and E. F. Camacho. MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9):2382–2387, 2008.

[87] D. Limon, M. Pereira, D. M. de la Peña, T. Alamo, C. N. Jones, and M. N. Zeilinger. MPC for tracking periodic references. *IEEE Transactions on Automatic Control*, 61(4):1123–1128, 2016.

[88] Y. Lu and Y. Arkun. A scheduling quasi-minmax MPC for LPV systems. In *American Control Conference, 1999. Proceedings of the 1999*, volume 4, pages 2272–2276. IEEE, 1999.

[89] A. I. Maalouf. Improving the robustness of a parallel robot using Predictive Functional Control (PFC) tools. In *Decision and Control, 2006 45th IEEE Conference on*, pages 6468–6473, 2006.

[90] J. M. Maciejowski. *Predictive control: with constraints*. Pearson education, 2002.

[91] U. Maeder, F. Borrelli, and M. Morari. Linear offset-free model predictive control. *Automatica*, 45(10):2214–2222, 2009.

[92] U. Maeder and M. Morari. Offset-free reference tracking with model predictive control. *Automatica*, 46(9):1469–1476, 2010.

[93] D. Mayne. An apologia for stabilising terminal conditions in model predictive control. *International Journal of Control*, 86(11):2090–2095, 2013.

[94] D. Q. Mayne, S. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222, 2006.

[95] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[96] D. Q. Mayne, M. M. Seron, and S. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[97] B. Mettler. *Identification modeling and characteristics of miniature rotorcraft*. Springer Science & Business Media, 2013.

[98] R. H. Middleton, J. Chen, and J. S. Freudenberg. Tracking sensitivity and achievable H_∞ performance in preview control. *Automatica*, 40(8):1297–1306, 2004.

[99] A. Morshedi, C. Cutler, and T. Skrovanek. Optimal solution of dynamic matrix control with linear programing techniques (LDMC). In *American Control Conference, 1985*, pages 199–208, 1985.

[100] K. R. Muske and T. A. Badgwell. Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5):617–632, 2002.

[101] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[102] A. O'Dwyer. *Handbook of PI and PID controller tuning rules*, volume 57. World Scientific, 2009.

[103] B. A. Ogunnaike. Dynamic matrix control: A nonstochastic, industrial process control technique with parallels in applied statistics. *Industrial & Engineering Chemistry Fundamentals*, 25(4):712–718, 1986.

[104] A. W. Ordys and P. Kock. Constrained predictive control for multivariable systems with application to power systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 9(11):781–797, 1999.

[105] G. Pannocchia. Robust model predictive control with guaranteed setpoint tracking. *Journal of Process control*, 14(8):927–937, 2004.

[106] G. Pannocchia. Offset-free tracking mpc: A tutorial review and comparison of different formulations. In *Control Conference (ECC), 2015 European*, pages 527–532, 2015.

[107] G. Pannocchia and A. Bemporad. Combined design of disturbance model and observer for offset-free model predictive control. *IEEE Transactions on Automatic Control*, 52(6):1048–1053, 2007.

[108] G. Pannocchia and E. C. Kerrigan. Offset-free receding horizon control of constrained linear systems. *AIChE Journal*, 51(12):3134–3146, 2005.

[109] G. Pannocchia and J. B. Rawlings. The velocity algorithm LQR: a survey. *Technical Report 2001-01, TWMCC*, 2001.

[110] G. Pannocchia and J. B. Rawlings. Disturbance models for offset-free model-predictive control. *AIChE journal*, 49(2):426–437, 2003.

[111] G. Pannochia. *Strategies for robust multivariable model-based control*. PhD thesis, PhD Thesis, University of Pisa, 2002.

[112] E. N. Pistikopoulos, V. Dua, N. A. Bozinis, A. Bemporad, and M. Morari. On-line optimization via off-line parametric optimization tools. *Computers & Chemical Engineering*, 24(2-7):183–188, 2000.

[113] B. Pluymers, J. Rossiter, J. Suykens, and B. De Moor. A simple algorithm for robust MPC. In *IFAC World Congress*, 2005.

[114] B. Pluymers, J. Rossiter, J. Suykens, and B. De Moor. Efficient computation of polyhedral invariant sets for LPV systems and application to robust MPC. *Submitted for publication*, 2005.

[115] F. A. Potra and S. J. Wright. Interior-point methods. *Journal of Computational and Applied Mathematics*, 124(1-2):281–302, 2000.

[116] G. Prasath and J. B. Jørgensen. Soft constraints for robust MPC of uncertain systems. *IFAC Proceedings Volumes*, 42(11):225–230, 2009.

[117] G. Prasath, B. Recke, M. Chidambaram, and J. B. Jørgensen. Application of soft constrained MPC to a cement mill circuit. *IFAC Proceedings Volumes*, 43(5):302–307, 2010.

[118] D. M. Prett and R. Gillette. Optimization and constrained multivariable control of a catalytic cracking unit. In *Joint Automatic Control Conference*, number 17, 1980.

[119] S. J. Qin and T. A. Badgwell. An overview of industrial model predictive control technology. In *AIChE Symposium Series*, volume 93, pages 232–256, 1997.

[120] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[121] M. R. Rajamani and J. B. Rawlings. Estimation of the disturbance structure from data using semidefinite programming and optimal weighting. *Automatica*, 45(1):142–148, 2009.

[122] M. R. Rajamani, J. B. Rawlings, and S. J. Qin. Achieving state estimation equivalence for misassigned disturbances in offset-free model predictive control. *AIChE Journal*, 55(2):396–407, 2009.

[123] S. Rakovic and D. Mayne. A simple tube controller for efficient robust model predictive control of constrained linear discrete time systems subject to bounded disturbances. In *IFAC World Congress*, 2005.

[124] A. Ramdani and S. Grouni. Dynamic matrix control and generalized predictive control, comparison study with IMC-PID. *International Journal of Hydrogen Energy*, 42(28):17561–17570, 2017.

[125] C. V. Rao and J. B. Rawlings. Steady states and constraints in model predictive control. *AIChE Journal*, 45(6):1266–1278, 1999.

[126] J. B. Rawlings, D. Bonné, J. B. Jørgensen, A. N. Venkat, and S. B. Jørgensen. Unreachable setpoints in model predictive control. *Automatic Control, IEEE Transactions on*, 53(9):2209–2215, 2008.

[127] J. B. Rawlings and D. Q. Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[128] J. B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. *IEEE transactions on automatic control*, 38(10):1512–1516, 1993.

[129] J. Richalet. Industrial applications of model based predictive control. *Automatica*, 29(5):1251–1274, 1993.

[130] J. Richalet, E. Abu, C. Arber, H. Kuntze, A. Jacubasch, and W. Schill. Predictive functional control: application to fast and accurate robots. In *Proceedings of the 10th IFAC Congress*, volume 4, pages 251–258, 1987.

[131] J. Richalet, T. Darure, and J. Mallet. Predictive functional control of counter current heat exchangers. *IFAC Proceedings Volumes*, 47(3):5345–5350, 2014.

[132] J. Richalet and D. O'Donovan. *Predictive functional control: principles and industrial applications*. Springer Science & Business Media, 2009.

[133] J. Richalet, A. Rault, J. Testud, and J. Papon. Model predictive heuristic control. *Automatica (Journal of IFAC)*, 14(5):413–428, 1978.

[134] A. Richards. Fast model predictive control with soft constraints. In *Control Conference (ECC), 2013 European*, pages 1–6, 2013.

[135] A. Richards. Fast model predictive control with soft constraints. *European Journal of Control*, 25:51–59, 2015.

[136] P. Roberts. A brief overview of model predictive control. In *Practical Experiences with Predictiv, IEE Seminar on*, pages 1–1, 2000.

[137] J. Rossiter. A global approach to feasibility in linear MPC. *Proc. UKACC ICC*, 2006.

[138] J. Rossiter, J. Gossner, and B. Kouvaritakis. Infinite horizon stable predictive control. *IEEE transactions on automatic control*, 41(10):1522–1527, 1996.

[139] J. Rossiter and B. Grinnell. Improving the tracking of generalized predictive control controllers. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 210(3):169–182, 1996.

[140] J. Rossiter and B. Kouvaritakis. Constrained stable generalised predictive control. In *IEE Proceedings D-Control Theory and Applications*, volume 140, pages 243–254, 1993.

[141] J. Rossiter, B. Kouvaritakis, and M. Cannon. An algorithm for reducing complexity in parametric predictive control. *International Journal of Control*, 78(18):1511–1520, 2005.

[142] J. Rossiter, B. Kouvaritakis, and R. Dunnett. Application of generalised predictive control to a boiler-turbine unit for electricity generation. In *IEE Proceedings D (Control Theory and Applications)*, volume 138, pages 59–67, 1991.

[143] J. Rossiter, B. Kouvaritakis, and J. R. Gossner. Guaranteeing feasibility in constrained stable generalised predictive control. *IEE Proceedings-Control Theory and Applications*, 143(5):463–469, 1996.

[144] J. Rossiter and G. Valencia-Palomo. Feed forward design in MPC. In *Control Conference (ECC), 2009 European*, pages 737–742, 2009.

[145] J. A. Rossiter. *Model-based predictive control: a practical approach*. CRC press, 2003.

[146] J. A. Rossiter, B. Kouvaritakis, and M. Rice. A numerically robust state-space approach to stable-predictive control strategies. *Automatica*, 34(1):65–73, 1998.

[147] J. Rovnak and R. Corlis. Dynamic matrix based control of fossil power plants. *Energy Conversion, IEEE Transactions on*, 6(2):320–326, 1991.

[148] V. Sakizlis, V. Dua, J. D. Perkins, and E. N. Pistikopoulos. The explicit control law for hybrid systems via parametric programming. In *American Control Conference, 2002. Proceedings of the 2002*, volume 1, pages 674–679, 2002.

[149] V. Sakizlis, K. I. Kouramas, and E. N. Pistikopoulos. Linear model predictive control via multiparametric programing. *Book: Process Systems Engineering*, 2, 2007.

[150] V. Sakizlis, J. D. Perkins, and E. N. Pistikopoulos. The explicit model-based tracking control law via parametric programming. *IFAC Proceedings Volumes*, 37(1):631–636, 2004.

[151] T. L. Santos. Modified reference tracking MPC for constrained linear systems. *International Journal of Systems Science*, pages 1–11, 2018.

[152] P. O. Scokaert and J. B. Rawlings. Infinite horizon linear quadratic control with constraints. *IFAC Proceedings Volumes*, 29(1):5905–5910, 1996.

[153] P. O. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *Automatic Control, IEEE Transactions on*, 43(8):1163–1169, 1998.

[154] P. O. Scokaert and J. B. Rawlings. Feasibility issues in linear model predictive control. *AIChE Journal*, 45(8):1649–1659, 1999.

[155] M. Shahrokhi and A. Zomorrodi. Comparison of PID controller tuning methods. *Department of Chemical & Petroleum Engineering Sharif University of Technology*, pages 1–2, 2013.

[156] L. Shead, K. Muske, and J. Rossiter. Conditions for which MPC fails to converge to the correct target. In *IFAC World Congress*, pages 6968–6973, 2008.

[157] L. Shead, K. Muske, and J. Rossiter. Conditions for which linear MPC converges to the correct target. *Journal of Process Control*, 20(10):1243–1251, 2010.

[158] L. Shead and J. Rossiter. Feasibility for non-square linear MPC. In *ACC*, pages 4464–4469, 2007.

[159] D. Simon. *Model Predictive Control in Flight Control Design: Stability and Reference Tracking*. PhD thesis, Linköping University Electronic Press, 2014.

[160] M. Sznaier and M. J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *Decision and Control. 26th IEEE Conference on*, volume 26, pages 761–762, 1987.

[161] P. Tatjewski. Disturbance modeling and state estimation for offset-free predictive control with state-space process models. *International Journal of Applied Mathematics and Computer Science*, 24(2):313–323, 2014.

[162] P. Tatjewski. Offset-free nonlinear model predictive control with state-space process models. *Archives of Control Sciences*, 27(4):595–615, 2017.

[163] P. TøNdel, T. A. Johansen, and A. Bemporad. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3):489–497, 2003.

[164] T. Tsang and D. W. Clarke. Generalised predictive control with input constraints. In *IEE Proceedings D (Control Theory and Applications)*, volume 135, pages 451–460, 1988.

[165] G. Valencia-Palomo. *Efficient implementations of predictive control*. PhD thesis, University of Sheffield, 2010.

[166] G. Valencia-Palomo, J. Rossiter, and F. López-Estrada. Improving the feed-forward compensator in predictive control for setpoint tracking. *ISA transactions*, 53(3):755–766, 2014.

[167] L. Wang. *Model predictive control system design and implementation using MATLAB®*. springer, 2009.

[168] B. Wie and D. S. Bernstein. Benchmark problems for robust control design. *Journal of Guidance, Control, and Dynamics*, 15(5):1057–1059, 1992.

[169] A. Wills and W. Heath. An exterior/interior-point approach to infeasibility in model predictive control. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 4, pages 3701–3705, 2003.

[170] E. L. S. Wong. *Active-set methods for quadratic programming*. PhD thesis, UC San Diego, 2011.

[171] C. Xia, M. Wang, Z. Song, and T. Liu. Robust model predictive current control of three-phase voltage source PWM rectifier with online disturbance observation. *IEEE Transactions on Industrial Informatics*, 8(3):459–471, 2012.

[172] E. Zafiriou and H.-W. Chiou. Output constraint softening for SISO model predictive control. In *American Control Conference, 1993*, pages 372–376, 1993.

[173] M. N. Zeilinger, M. Morari, and C. N. Jones. Soft constrained model predictive control with robust stability guarantees. *IEEE Transactions on Automatic Control*, 59(5):1190–1202, 2014.

[174] R. Zhang, S. Wu, and F. Gao. State space model predictive control for advanced process operation: A review of recent development, new results, and insight. *Industrial & Engineering Chemistry Research*, 56(18):5360–5394, 2017.

[175] A. Zheng and M. Morari. Stability of model predictive control with mixed constraints. *IEEE Transactions on Automatic Control*, 40(10):1818–1823, 1995.

Appendix A

CREATING RANDOM SYSTEMS

This appendix presents the procedures required to create three different random systems (over damped, critical damped and unstable systems) as follows.

A.1 Defining random eigenvalues

For n^{th} order systems, there must be n eigenvalues $(\lambda_1, \lambda_2, \cdots, \lambda_n)$, these values can be determined as required, using the following formulas.

$$\lambda_i = (b_i - a_i)X + a_i, \qquad i = 1, 2, 3, \cdots, n. \tag{A.1}$$

where X is a random variable drawn independently from a standard uniform distribution with values in the interval (0,1) and $n,\ b_i,\ a_i$ are the number and upper/lower limits of the eigenvalues respectively.

A.2 Creating random system matrices

In order to create a random n^{th} order system, we need to create random system matrices $A,\ B$, and C as follows.

A.2.1 The matrix A

The system matrix A is given by:

$$A = M\Lambda M^{-1} \tag{A.2}$$

where M and Λ are random matrices with elements drawn independently from distribution A.1.

The eigenvalues are chosen such that:

- For over damped systems, all the eigenvalues are different, and must satisfy this constraint; $0 \leq \lambda_i \leq 1$.
- For critical damped systems, all the eigenvalues are equal real values, and must satisfy this constraint; $0 \leq \lambda_i \leq 1$.
- For unstable systems, at least one eigenvalue must be greater than 1.

A.2.2 The matrix B

The system matrix B is given by:

$$B = V$$

where V is a random matrix with elements drawn independently from distribution X. The controllability of the system can be tested through computing the controllability matrix.

$$\mathbb{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{A.3}$$

The system is controllable, if the controllability matrix $\mathbb{C}$ must has a full rank [7].

A.2.3 The matrix C

The system matrix C is given by:

$$C = W$$

where W is a random matrix with elements drawn independently from distribution X. The observability of the system is tested through the observability matrix.

$$\mathbb{Q} = \begin{bmatrix} C \\ AC \\ A^2C \\ \vdots \\ A^{n-1}C \end{bmatrix} \tag{A.4}$$

The system is observable, if the observability matrix $\mathbb{Q}$ has a full rank [7].

Appendix B

MODEL STATE SPACE PARAMETERS

This appendix presents model matrices, A, B, C and D, which describe the process model of both the column two gas treatments of (8.14) and the compressor of (4.21). These matrices are determined by converting the transfer matrix to state space formulation using a $(tf2ss)$ function in MATLAB 2017a. The obtained matrices are described in the following sections:

B.1 Compressor model (4.21)

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{B.1}$$

where

$$
A_{11} = \left[\begin{array}{cccccccccccccccccccccccccccccc}
1.881 & -0.882 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.8534 & -0.8589 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}\right]
\tag{B.2}
$$

$$A_{12} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{B.3}$$

$$A_{21} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \tag{B.4}$$

$$
A_{22} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.9761 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.2919 & -0.3060 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} \tag{B.5}
$$

$$
B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \tag{B.6}
$$

where

$$B_1 = \begin{bmatrix} 0.0156 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0625 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{B.7}$$

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.125 & 0 & 0.0625 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{B.8}$$

$$C = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \tag{B.9}$$

where

$$C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0049 & 0.0047 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0178 & 0.0169 & 0 & 0 \end{bmatrix} \tag{B.10}$$

$$C_2 = \begin{bmatrix} 0 & 0 & 0.1760 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0431 & -0.0291 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{B.11}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{B.12}$$

B.2 Column two gas-plant model (8.14)

$$
A = \begin{bmatrix}
0.9683 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.8948 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9429 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.9260 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix} \tag{B.13}
$$

$$
B = \begin{bmatrix}
0.5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}^T \tag{B.14}
$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & -0.3810 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.9140 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5258 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.1850 & 0 \end{bmatrix} \tag{B.15}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{B.16}$$