The University Of Sheffield.

Efficient parameterise solutions of predictive control

Bilal Khan

Thesis submitted in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy

Automatic Control and Systems Engineering
University of Sheffield
June 2013

Abstract

Model based predictive control (MPC) is well established and has gained widespread acceptance in the industry and the academic community. The success of earlier industrial heuristic MPC algorithms motivated the research community to develop several algorithms with improved performance and enlarge the region of attraction. All proposed algorithms to some extent form a trade off between a region of attraction, performance and inexpensive optimisation. This thesis makes contributions in the area of MPC algorithm design and in particular examines to what extent different methods for parameterising the degrees of freedom within the input trajectories can improve aspects of the region of attraction, performance and inexpensive optimisation.

Kautz functions are explored to parameterise the input sequences in optimal model predictive control (OMPC). It is shown that this modification gives mechanisms to achieve low computational burden with enlarged region of attraction and without too much detriment to performance. The proposed algorithm based on Kautz function parameterisation guarantees stability and recursive feasibility. It is further explored and a general class of function parameterisation is proposed using higher order orthonormal basis functions. A generalised function based MPC algorithm is formulated with guaranteed convergence and recursive feasibility. The efficacy of the proposed parameterisations within existing MPC algorithms are demonstrated by examples.

The general class of function parameterisation is further explored by looking at systematic choices for a particular problem. Systematic mechanisms are discussed to choose the best tuned alternative parameterisation dynamics. The numerical examples demonstrate the efficacy of the systematic mechanisms. It is also shown that generalised function parameterisations are computationally efficient when used to achieve an approximately global region of attraction as compared with OMPC, there is a reduction in number of inequalities to represent the region of attraction, the number of multiparametric solutions (and therefore computational complexity and memory storage) and also the computa-

tional time using active set methods.

Another avenue explored is the efficacy of generalised function parameterisation of the degree of freedom within a robust MPC algorithm. It extends the work of nominal case to the robust scenario and shows that similar benefits accrue. An algorithm is proposed for the robust MPC using the generalised function parameterisation that enables the use of robust control invariant set to enlarge the region of attraction.

Finally, the parameterised solution extends to triple mode approaches to simplify the offline computations. In triple mode MPC algorithms, the first novelty is to propose explicit choices of middle mode using generalised function dynamics as a pragmatic choice without demanding offline computations. The second novel contribution is to parameterise the input sequences for both explicit and implicit choices of the middle mode within triple mode MPC algorithms. The improvements, with respect to existing algorithms, are demonstrated by examples.

Dedicated to my Parents

Acknowledgements

First and foremost I am thankful to Almighty ALLAH, without His blessings it is impossible to complete this thesis. My humblest and heartfelt Darood to Holy Prophet MUHAMMAD (Peace Be Upon Him), who educated his followers to seek the knowledge from cradle to grave.

I would like to express my deepest sense of gratitude to my supervisor, Dr J. Anthony Rossiter, for his profound knowledge, masterly creative thinking, and consistent encouragement, which was a source of inspiration through the course of this work. Thank you Anthony for giving me the opportunity to conduct this research and for constant encouragement during my studies. Your gentle personality and positive attitude in research will benefit my career as well as personal life.

I would like to express my thanks to Guillermo Valencia-Palomo (Inst. Tecnol. de Hermosillo, Mexico) and Mark Cannon (University of Oxford,UK) for their collaboration at some point of my research. I would like to thank Saam for useful discussions. Special thanks to my friends Anwar, Majid, Nisar and as well as many others sharing good times during my stay in Sheffield.

I would have never reached this point in my life without the support of my parents, from the beginning of my education all the way through to the complication of my PhD studies. I am grateful for the sacrifice you made and for creating the environment that has encouraged me to succeed. Of course this environment includes my brother, my sisters, my nephews and my niece to whom I am grateful as well. I am also thankful to my fiancee, a lovely addition to my life during my PhD studies for her endless love, always being there for me and making a safe-haven for all types of stress; you are my joy, my piece, my cornerstone.

Finally, I extend a special thanks to my sponsors, University of Sheffield ,UK and COMSATS Institute of Information Technology, Pakistan for funding my studies and believing in me.

Contents

List of Figures

List of Tables

Nomenclature

Upper-case roman letters

A, B, C	Model matrices: state transition, input and output matrices respectively.
A_L, A_K, A_G	Laguerre, Kautz and generalised functions prediction matrices.
D, E	Perturbation transformation matrices.
G_i, L_i	The ith discrete-time generalised and Kautz function.
G_c, H	Dynamic matrix for perturbation.
H_G, H_K, H_L	Matrices for discrete time generalised, Kautz and Laguerre functions.
I_{n_c}	$n_u \times n_u$ identity matrix.
J, J_c, J_L, J_K, J_G	Cost functions.
K	Feedback gain matrix.
L_u, L_x	Matrices for inequality constraints.
$M, N, N_1, N_2, \hat{N}$	Matrices for repesenting MCAS.
M_G, M_K, N_G, N_K	Matrices for representing MCAS.
M_T, N_T	Matrices for triple mode MCAS.
M_r, N_r	Matrices for robust MCAS.
$P_{cl}, H_c, P_{clu}, H_{cu}$	Prediction matrices.
P_e, Q_z	Matrices for ellipsoidal set.
$P, \tilde{P}$	Penalty matrices for triple mode cost.
P, h	Mann-Whitney U test parameters in Chapter 6
Q, Q_f	State penalty matrix.
R	Input penalty matrix.
S, S_c, S_G, S_L, S_η	Penalty matrices for perturbation to state-back control law.
S_T, W_q, W_1	Penalty matrices for perturbation to state feedback control law.
T	Linear Transformation matrix.
O_{n_u}	$n_u \times n_u$ zero matrix.

Lower-case roman letters

$a, b, a_1 ... a_n, p$	Scaling factors for Kautz, *nth* order and Laguerre functions.
c, f, q	Perturbation to state-feedback control law.
$g_i(.), k_i(.), l_i(.)$	The *ith* discrete time generalised function, Kautz and Laguerre functions in vector form.
n_c, n_q, n_G, m_c	Perturbation control horizons.
e_i	*ith* standard basis vector.
r	Setpoint.
u, x, y, z	Vector of system inputs, states, outputs and controlled variables.
k	Time-step index.
n_u, n_x, n_y	Dimensions for the system input, output and state vectors.
$d, d_T, d_r, \tilde{d}_r$	Vectors used for MCAS.

Greek letters

$\sigma, \lambda, \gamma, \zeta_i$	Scale factors.
α	Tuning variable (Chapter 6).
β	Average closed loop performance.
η, γ, ρ	Laguerre, Kautz and generalised function perturbations respectively.
λ	Lagrange multiplier vector.
Φ	Closed loop transition matrix.
Ψ, ψ	Augmented dynamics.
$\Phi_i, \Psi_i, \tilde{\Psi}_i, \psi_i, \tilde{\psi}_i$	The *ith* closed loop transition and augmented dynamics.
Γ	Penalty matrices for perturbation to state-back control law.
ν	Average MCAS volume.

Special characters

$\mathcal{A}, \mathcal{P}$	Polytopic sets.
$\mathcal{E}_x, \mathcal{E}_z, \mathcal{X}_E$	Ellipsoid sets.
$\mathcal{K}_i$	The *ith* discrete-time Kautz function.
$\mathbb{N}$	Set of natural numbers.

$\mathbb{R}^i$	Set of real numbers of dimension i.
$\mathbb{U}, \mathbb{X}, \mathbb{Y}$	Inputs, states and outputs linear inequalities.
$\mathcal{W}_k$	Working set of ASM.
$\mathcal{X}$	Polyhedral set for inequalities.
$\mathcal{X}_c, \mathcal{X}_r, \mathcal{X}_{r_g}$	Polyhedral set for MCAS and robust MCAS.
$\mathcal{X}_0, \mathcal{X}_{0,r}$	Polyhedral set for MAS and RMAS.

Superscripts

$*$	Denoted optimality.
$-$	Upper bound of a variable.

Subscripts

$\{i\}$	ith row selection of a matrix.
$k\|i$	Value of a variable at time step k evaluated at time step i.
max	Maximum bound of a variable.
min	Minimum bound of a variable.
ss	Steady-state value of a variable.
$-$	Lower bound of a variable.
$\rightarrow$	Column vector of prediction.

Functions

$Co\{.\}$	Convex hull.
$diag(\times)$	Block diagonal matrix of $\times$ matrices.
$tr(.)$	Trace of a matrix.
$vol(.)$	Volume of a polyhedral set.

Acronyms and abbreviations

ASM	Active Set Method.
BMI	Bilinear Matrix Inequality.
CARIMA	Controlled Auto Regressive Integrated Moving Average.
DMC	Dynamic Matrix Control.
d.o.f.	Degree of freedom.
EPs	Equalities Problems.
ERPC	Efficient Robust Predictive Control.
GERPC	Generalised ERPC.
GIMPC	Generalised Interpolation MPC.
GOMPC	Generalised OMPC.
GPC	Generalised Predictive Control.
GR(E)TMPC	Generalised function based RTMPC using an ellipsoidal set.
GRTMPC	Generalised function based RTMPC.
IDCOM	Identification and Command.
KKT	Karush-Kuhn-Tucker.
KOMPC	Kautz OMPC.
KRTMPC	Kautz based robust triple mode MPC.
LMI	Linear Matrix Equality.
LOMPC	Laguerre OMPC.
LP	Linear Programming.
LPV	Linear parameter varying.
LQR	Linear Quadratic Regulator.
LR(E)TMPC	Laguerre robust ellipsoid based triple mode MPC.
LRTMPC	Laguerre based RTMPC.
LTV	Lineat Time Varying.
MAC	Model Algorithmic Control.
MAS	Maximum Admissible Set.
MCAS	Maximum Controllable Admissible Set.
MIMO	Multiple Inputs, Multiple Outputs.
MPC	Model (based) Predictive Control.
mpGOMPC	GOMPC using muti-parametric solutions.
mpKOMPC	KOMPC using muti-parametric solutions.
mpLOMPC	LOMPC using muti-parametric solutions.
MPT	Mutli-parametric Toolbox.
NMPC	Non-linear MPC

NSG-II	Non-dominated Sorting Genetic Algorithm-II.
OMPC	Optimal MPC.
PFC	Predictive Functional Control.
PID	Proportional Integral Derivative.
PWA	Piecewise affine.
QDMC	Quadratic Dynamic Matrix Control.
QP	Quadratic Program.
SDP	Semidefinite Programming.
SMOC	Shell multivariable optimising control.
SISO	Single Input, Single Output.
s.t.	Subject to.
RGMPC	Robust generalised MPC.
RHC	Receding Horizon Control.
RMAS	Robust MAS
RMPC	Robust MPC.
RTMPC	Robust triple mode MPC.

Chapter 1

Introduction

Today, the process industry is characterised by product quality specifications which become more and more tight, increasing productivity demands, new environmental regulations and fast changes in the economic market. As a consequence process industry is nowadays confronting a strongly competitive environment and extracting greater value from manufacturing assets is a major challenge. One of the main reasons of all these changes is the globalisation of the market.

Nowadays, companies seeking to increase profitability are shifting from a supplier driven market to customer-centric, demand-driven manufacturing environments where product quality and customer service is becoming essential for success. In many cases the requirements are contradictory for example, there is always a constant push towards a higher quality of products with lower manufacturing costs [1]. In this context, nowadays, the role of the engineer is to design, within budget and available time, a controller which is guaranteed to meet the client's specifications subject to energy costs, environmental and safety demands in the presence of changes in the characteristics of the process and variable demands.

The rapid development of control technology has an impact on the process industry. New theory, new controllers, actuators, sensors, new industrial process, computer methods, new applications, new philosophies, new challenges are proposed in the area of control discipline. The control technology offers a potential to implement more advanced control algorithms but often the preferred strategy of many industrial engineers is to design a robust and transparent process control structure that uses simple controllers. This is one reason why the PID controller remains industry's most implemented controller; however, this approach can create limitations on the process efficiency [2]. One such limitation is

the possible lack of a systematically achieved performance within the process hierarchy. For example in the case of reference tracking, PID control might be too short- sighted for the tracking performance. Another is the omission of a facility to accommodate and handle process operational constraints.

All industrial systems have inputs and outputs which are limited in size due to the presence of safety or physical constraints. Furthermore, an industrial process design might also require a certain level of performance, which can be translated into additional constraints on the controlled system. Excluding these constraints in the controller design phase may lead to a control action that could result in constraint violations. Depending on the underlying application, a violation might result in system failure, which in turn could possibly become a human hazard.

The current interest in the industry due to the emergence of advanced control techniques provides a great opportunity to improve process efficiency and optimality in the presence of constraints. Advanced control includes a vast number of methods which provide important ways in which the production situation can be improved; model based predictive control is one of the most successful solutions for an appropriate operation.

1.1 Model Predictive Control

Model (Based) Predictive Control (MPC) or receding horizon control (RHC) or moving horizon or embedded optimisation or real time optimisation or predictive control [3–6], are general names for different computer control algorithms that use past information of the inputs and outputs and a mathematical model of the plant to optimise its predicted future behaviour. During the last few decades, MPC has become a leading industrial control technology mainly due to the ease with which constraints can be included in the controller formulation.

MPC approaches determine a sequence of optimal control actions (inputs) over a future time horizon in order to optimise the performance of the controlled system, expressed in terms of a cost function. The optimisation is based on an internal mathematical model which, given the current measurements, predict the future behaviour of the real system with respect to changes in the control inputs. Once the sequence of optimal control inputs has been determined, only the first element is actually implemented and the optimisation is repeated at the next time interval with the new measurements and over the shifted horizon. This feedback mechanism of the MPC compensates for the prediction error due to structural mismatch between the internal model and the real system as well as for disturbances.

The main advantages which make MPC industrially desirable are that it can address the control problem with constrained optimisation.

- The possibility to express constraints explicitly in the problem formulation offers a natural way to state a broad class of control problems.
- Often the best performance, which may correspond to the most efficient or profitable operation, is obtained when the system is made to operate near the constraints.
- In the presence of actuator saturations, a control approach that is aware of the constraints never generates control inputs beyond the saturation values, and this removes the wind-up problem.

In addition, MPC approaches have the advantage of naturally handling multivariable control problems and systems with complex dynamics (like systems with time delays, for example). MPC approaches are powerful and robust in comparison with standard PID control, and their relative ease to configure and tune allows a remarkably short pay back time [7–9].

The basic principle of MPC is illustrated in Figure 1.1 where a single input single output system is considered. At each sampling time k, finite horizon optimal control problem is solved over a prediction horizon, using the current state of the process as the initial state. The output is required to follow a set point r. The figure also gives the previous history of the output trajectory and of the implied input at time instant k, which is subject to a saturation constraint. The online optimisation problem takes account of system dynamics, constraints and control objectives. The optimisation yields an optimal control sequence (represented as control horizon in Figure 1.1), but only the control action for the current time is applied while the rest of the calculated sequence is discarded. At the next time instant the horizon is shifted one sample and the optimisation is restarted with the information of the new measurements, using the concept of the receding horizon.

1.2 Motivation

MPC is well established and widely used both in the process industry and the control research community, but nevertheless there are still some theoretical and practical issues which have non-satisfactory answers. MPC algorithms have been successfully applied to many real systems [9], including multivariable systems, because it builds constraint

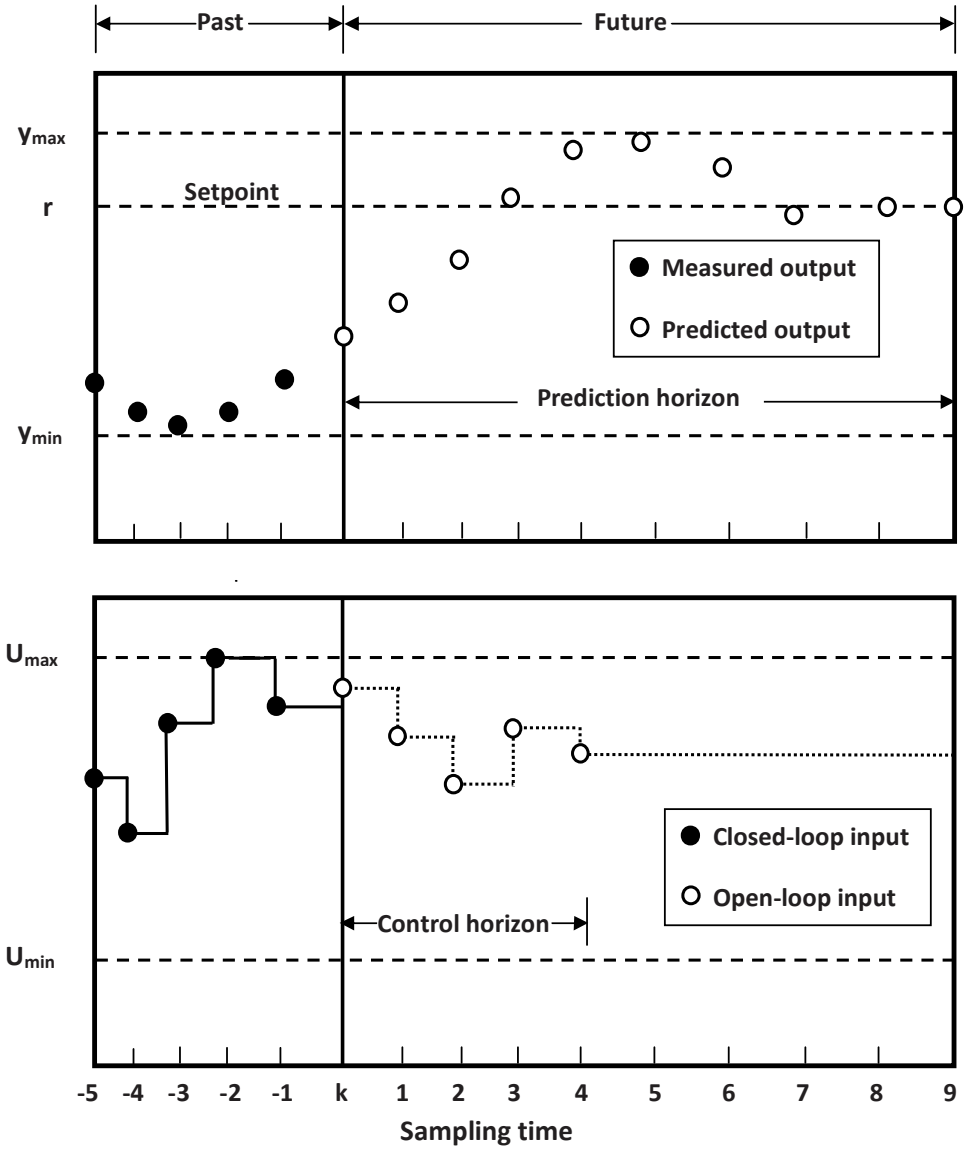

Figure 1.1: Graphical illustration of Model Predictive Control strategy

handling in from the outset. There are five aspects of MPC algorithm design, which have direct impact on the performance and complexity of the controller implementation. It will become clear that some of these goals are conflicting, which will illustrate the need for improved algorithms and this forms the main motivation for this thesis. The challenge relies on choosing the best balance according to the application needs.

1. **Stability**

   One of the most important aims is obviously to obtain a controller that stabilises

the underlying system according to some stability measure. MPC control design includes the notion of stability [6, 10, 11]. A common requirement is to guarantee stability in a sense relevant to the control objectives, i.e. stabilisation, tracking or disturbance rejection.

2. **Region of attraction**
   An issue related to stability is the region of attraction. It is defined as the region of the state space for which the system is stable. In MPC algorithm design, there is a need for the algorithm to be feasible, at every sampling instant. This requirement is called recursive feasibility. Within MPC, feasibility should really be defined as recursive feasibility. Typically, the region of attraction is also region for which stability is guaranteed. Consequently, another important aim is to obtain a region of attraction that is as large as possible. However, in combination with ensuring stability [6, 10, 11], the region of attraction may be small unless one uses a prohibitively large number of control moves (or degrees of freedom).

3. **Performance**
   One of the major selling points of MPC is its ability to do online constraint handling in a systematic way, hopefully retaining to some extent the stability margins and performance subject to constraint satisfaction. MPC controllers are used when the constraints play a determinant role in the control problem and as such, typically, improves the obtainable control performance significantly compared to the unconstrained case. However, when operating in regions away from the constraints, it is desirable that the MPC controller closely achieves an optimal unconstrained controller behaviour e.g. Linear Quadratic Regulator or LQR controller. Typically optimal tuned terminal control laws (e.g. the LQR optimal) lead to a relatively conservative region of attraction, whereas large regions of attraction typically correspond to suboptimal terminal controllers. Another way of improving optimality is increasing the degrees of freedom.

4. **Computational Complexity**
   An important issue for real time implementation of MPC is to find the fastest ways of optimisation as the time required for solving online optimisation may be very limited. Therefore, the computational efficiency of an algorithm in this aspect becomes critical when the algorithm is designed. The computational complexity is mainly determined by the number of optimisation variables, the number of constraints, the length of the horizon and the class of optimisation problems. Therefore, a trade off has to be made between performance, region of attraction and the computational burden when choosing from the currently available algorithms. Due to the computationally expensive online optimisation which is required, historically there has

been some limitation to the applications on which MPC has been used.

5. **Robustness**
   Finally, another important aim of MPC controller design is its robustness with respect to differences between the prediction model and the plant model and robustness with respect to external disturbances that act upon the system. If the effect of the uncertainty in the model is not taken into account, then the actual and theoretical behaviour of the system will differ. It is possible that a controller which does not take account of uncertainty would cause the control performance to be sluggish, overly conservative or in the worse scenario, would drive the system in an unsafe region. As with stability properties, adding mechanisms to improve robustness results in a typical trade off between computational complexity and region of attraction.

These design goals are conflicting and often not obtainable simultaneously, using classical MPC algorithms. Within the MPC community, there are now well accepted approaches which give the user confidence in the closed-loop performance and indeed it is easy to give a priori stability guarantee in the nominal case [6]. The most popular approach uses a dual mode prediction whereby one assumes some flexibility in the input predictions for some horizon, say n_c, and then assumes the input predictions follow a known fixed form in the longer horizon; usually this terminal assumption is equivalent to some fixed stabilising feedback law [5, 11]. The terminal assumption allows a straightforward definition of a feasible terminal set [12], that is a set within which the predictions are known to satisfy constraints. The overall region of attraction may be larger than this due to the control horizon n_c allowing some flexibility in transients and, in general up to some limit, one can grow the region of attraction by increasing the control horizon. Herein however lies a key trade off [13]:

- If the terminal law is well tuned, the terminal region may be small and hence a large n_c is required to get a large region of attraction. A large n_c could imply a large computational load and/or a more complex optimisation.
- If the terminal law is less well tuned, one may get a large region of attraction with a relatively small n_c, however one has embedded into the predictions a significant portion which is based on suboptimal dynamics and thus this will impact on the achievable performance.

If the user wishes to use the standard optimal MPC (OMPC) algorithm [5, 11] then the trade off is not easy to handle because the optimal control law is embedded as the

terminal law and therefore a relatively large n_c is required; however one could argue that at least the associated problem has a regular structure.

Some recent work has looked at very different prediction structures based on interpolation between two different strategies, say one with improved performance and one which enlarges the region of attraction e.g. [14–16] and other possible parameterisation approaches also exist [17–19]. Also, some authors have proposed blocking as means of extending the control horizon without increasing the number of d.o.f. [20]. However, this thesis does not pursue any of these approaches, instead, this thesis proposes to stick with a more conventional MPC structure, that is one based on a single set of predictions and a single terminal control law as for these, guarantees of stability and recursive feasibility are 'automatic' and asks whether alternative ways of selecting the input prediction profiles is beneficial. The specific constraint adopted in this thesis which facilities convergence guarantees is that the prediction profiles have a recursive property that within the prediction class, one can automatically choose the same profile as was selected at the previous sample. On the other hand the paper [21, 22], uses ellipsoidal constraints approximation to formulate an MPC Law, which provides robustness to model uncertainty. This type of approach was extended in [23] and [24] by allowing the parameters of the dynamic feedback law to be variables in the offline optimisation. These formulations can reduce the suboptimality in the size of stabilisable set associated with the MPC law of [21, 22], but leads to a nonconvex optimisation formulation with no guarantee of convergence to a solution. In [25], a convex formulation of the optimisation of prediction dynamics is proposed to enlarge the region of attraction using as highly tuned a terminal control law as is possible in combination with any detuned law. The proposed optimisation formulation simplifies the trade off between a region of attraction, performance and computational burden with following limitations:

- The proposed optimisation formulation is only convex for a prediction horizon which is equal (or exceeds) the system dimension.
- It was shown that there is no further gain in the region of attraction when the prediction horizon exceeds the system dimension and thus gives artificially tight limits on the gain in the region of attraction.
- The proposal is based on ellipsoids and hence it may be conservative in volume for non-symmetric constraints (This is consistent with the fact that ellipsoidal invariant set algorithms work best with symmetric constraints).
- The convex formulation may increase the prediction horizon for higher order system dimensions and hence although the region of attraction may be enlarged this is at the expense of a greater computational burden.

Therefore, the trade off between region of attraction, performance and computational burden requires further study.

The focus of this thesis is on the choice of the d.o.f. in a conventional OMPC type of algorithm. Historically the d.o.f. have been taken to be the individual control moves within the control horizon, however this is noted as being quite inefficient when the control horizon is by necessity large to ensure feasibility. Hence, an alternative approach is to consider combinations of different sequences of moves, where these sequences could in principle be over a relatively short horizon or in fact as in this thesis, sequences which evolve over an infinite horizon. As a means of convincing the reader of this concept, engineering commonsense already tells us that a sequence based on the unconstrained optimal control is likely to be a good one [11]. A set of sequences known to have been effective in earlier studies [18, 26] are the Laguerre polynomials which are a special case of the generalised prediction framework was proposed in [25] and thus a logical route of further study was to investigate this class of functions in more detail. This thesis develops this concept further and investigates the efficiency of more flexible functions along with systematic design guidelines for choosing appropriate functions to overcome the trade offs within MPC design goals. The efficacy of the proposed algorithms is considered for both nominal and robust scenarios.

1.3 Aims and objectives

The main aim of this thesis is to make contributions in the area of predictive control design and in particular to examine to what extent different methods for parameterising the degrees of freedom within the input trajectories can improve aspects of feasibility, performance and optimisation complexity. In order to do this, the proposed alternatives must preserve the benefits of the conventional algorithms such as constraint handling and high performance.

Particular objectives

To achieve the main aim, a number of particular objectives can be listed:

1. Propose flexible functions as an effective alternative/modification to the standard basis set for parameterising the degrees of freedom within MPC.

2. Generalise the parameterisation approaches to predictive control based on more flexible and Laguerre functions in a format to simplify the trade off between the region of attraction, performance and inexpensive optimisation.

3. Propose a systematic mechanism to choose the best tuned flexible function parameterisation dynamics.

4. Investigate the computational efficiency of using generalised function parameterisations for implicit and explicit solutions to MPC.

5. Demonstrate the efficiency of more flexible parameterisations of the degrees of freedom within a robust MPC algorithm.

6. Propose a polyhedral robust invariant set of an augmented system using generalised function parameterisation.

7. Propose Laguerre and Kautz function dynamics as a pragmatic choice within a triple mode MPC to simplify the offline computations.

8. Propose more flexible function parameterisations within a triple mode approaches to significantly enlarge the region of attraction.

1.4 Supporting publications

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

Conference papers

1. **Khan, B.**, J. A. Rossiter and Valencia-Palomo (2011). Exploiting Kautz function to improve feasibility in MPC. *18th IFAC World Congress Milano*, Italy.

2. **Khan, B.** and J. A. Rossiter (2011). Triple mode MPC or Laguerre MPC: a comparison. *American Control Conference - ACC'11*,San Francisco, USA.

3. **Khan, B.** and J. A. Rossiter (2011). Computational Efficiency of Laguerre MPC using Active Set Method. *IASTED Intelligent Systems and Control (ISC)*, Cambridge, UK.

4. **Khan, B.** and J. A. Rossiter (2011). Generalised parameterisation for MPC. *IASTED Intelligent Systems and Control - ISC*, Cambridge, UK.

5. **Khan, B.** and J. A. Rossiter (2012). A Comparison of the Computational Efficiency of multi-parametric predictive control using Generalised Function Parameterisations. *8th IFAC Symposium on Advanced Control of Chemical Processes - ADChem*, Singapore.

6. **Khan, B.** and J. A. Rossiter (2012). A Comparison of the Computational Efficiency of Generalised Function MPC using Active Set Methods. *8th IFAC Symposium on Advanced Control of Chemical Processes - ADChem*, Singapore.

7. **Khan, B.** and J. A. Rossiter (2012). Robust MPC algorithms using alternative parameterisations. *United Kingdom Automatic Control Conference - UKACC*, Cardiff, UK.

8. **Khan, B.** and J. A. Rossiter (2012). A systematic selection of an alternative parameterisation for predictive control. *United Kingdom Automatic Control Conference - UKACC*, Cardiff, UK.

Journal paper

1. **Khan, B.** and J. A. Rossiter (2013). Alternative parameterisation within predictive control: a systematic selection. *accepted, International Journal of Control*.

1.5 Thesis overview

The present document consists of 10 chapters and 1 appendix divided into 3 parts. A summary of each of them is presented next.

Part I Background and Literature review

In **Chapter 2**, the theoretical foundation and a brief history of the MPC algorithms are presented. A brief overview of nonlinear MPC with possible real time implementations, weaknesses and an overview of robust MPC is also presented. Thereafter real time implementation challenges are discussed in detail for the explicit and implicit solutions of MPC with their strengthens and weaknesses. Subsequently, the goals and challenges of any MPC implementation are discussed, along with a review of the most efficient MPC algorithms relevant to this thesis to establish state of the art approaches.

Chapter 3, provides a common theoretical framework necessary for arguments in this thesis. It introduces the class of system descriptions considered, the class of linear and linear parameter varying system with constraints, which is probably the most important and widely used one in practice. An overview of the problem formulation and stability in MPC is given. The chapter thereafter discusses MPC algorithms. A dual mode paradigm with an optimal MPC is discussed in detail, then the Laguerre function parameterisation with an optimal MPC is given. Triple mode MPC is presented using both polyhedral and ellipsoidal sets. Finally, an overview of robust MPC is given.

Part II Generalised function parameterisations within model predictive control

This part presents the contributions of the thesis to linear time invariant systems. The first two chapters (i.e. Chapter 4 and 5) propose alternative algorithms to simplify the trade off between the region of attraction, performance and computational load within the MPC. The last two chapters (i.e. Chapter 6 and 7) in this part discuss systematic tuning and the computational analysis of the proposed algorithms.

In **Chapter 4**, Kautz functions are presented as an alternative way to parameterise the input predictions in dual mode MPC. It is shown that Laguerre functions are a special case of Kautz functions. Specifically, a simple but efficient MPC algorithm that uses Kautz functions give a mechanism to achieve a low computational burden with enlarged the region of attraction and without degrading the performance is presented. This modification indeed may be more effective than the already proposed Laguerre functions as they offer more variety in the key characteristics. It is also shown that the proposed algorithms have standard convergence and feasibility guarantees. The improvement, with respect to the existing algorithm that uses a Laguerre parameterisation, is demonstrated by numerical examples.

In **Chapter 5**, generalised orthonormal basis functions are proposed to generalise approaches to predictive control based on Laguerre and Kautz functions. Laguerre and Kautz functions are presented as a special case of generalised functions. This chapter provides a possible alternative using orthonormal functions like Laguerre and Kautz functions. An algorithm based on a generalised function is presented with standard convergence and feasibility guarantees. The efficacy of the proposed parameterisation within existing predictive control algorithms that use a similar strategy (e.g. Laguerre and Kautz functions), is demonstrated by numerical examples.

So far generalised functions (i.e. Laguerre, Kautz and higher order orthonormal func-

tions) have been shown to be effective within predictive control design, but without giving explicit design guidelines. **Chapter 6** extends that work by looking at systematic choices for generalised functions. Systematic mechanisms are discussed to choose the best tuned alternative parameterisation dynamics. The efficacy of the proposed parameterisation within existing predictive control algorithms are demonstrated by examples.

In **Chapter 7**, the aim is to consider the computational efficiency of generalised function parameterisation for implicit and explicit solutions to MPC. It is shown that the generalised function parameterisation still uses a standard quadratic programming optimisation problem, this does not have the regular structure that one would desire. This chapter considers the compact problem formulation rising from removing any redundant constraints. Extensive numerical simulations are presented to show the computational analysis of the proposed algorithms using multi-parametric quadratic programming and active set method.

Part III An Efficient Robust model predictive control using generalised function parameterisation

This part presents the second part of the contribution of the thesis for uncertain dynamics. Chapter 8 uses dual mode prediction and Chapter 9 uses triple mode prediction.

In **Chapter 8**, the efficacy of flexible function parameterisation is extended to robust MPC algorithms and shown that similar benefits accrue as in the nominal cases. It is shown that increases in complexity of the robust case as compared to the nominal case is much less than might be expected. There are two key contributions, firstly to propose a polyhedral robust invariant set using flexible function parameterisation and secondly to introduce a robust MPC algorithm based on the proposed polyhedral robust invariant set to enlarge the region of attraction. It is also shown that the proposed algorithm has standard convergence and feasibility guarantees. To finish the chapter, numerical examples demonstrate the efficacy of the proposed algorithm.

In **Chapter 9**, the efficacy of more flexible function parameterisation is extended to triple mode approaches. It is shown that more flexible function parameterisation may significantly enlarge the region of attraction. There are many cases where such an approach is an improvement on earlier work and simplifies offline computations. The proposed algorithms are based on both ellipsoidal and polyhedral invariant sets. It is also shown that proposed algorithms have standard convergence and feasibility guarantees. To finish the chapter, numerical examples demonstrate the efficacy of the proposed algorithms.

Part IV Conclusion and Future directions

Finally, in **Chapter 10**, the original contributions of the thesis are summarised and the overall conclusions are presented. At the end of the chapter, the proposed future work is discussed.

Part V Appendix

In **Appendix A**, complementary information of Chapter 4 and 5 is given.

Part I

Background and Literature review

Chapter 2

Literature review

This chapter presents the literature review of predictive control in order to identify the key issues. Intentionally, the presentation of ideas and concepts in this chapter is kept at a rather colloquial level, rather than being mathematically rigorous. The chapter starts reviewing the foundation of model predictive control in optimal control theory, this is presented in Section 2.1. Then, in Section 2.2, the first few model predictive control algorithms are summarised in a historical context. Section 2.3, discusses briefly the nonlinear formulation of predictive control with possible real time implementation weaknesses. An overview of robust model predictive control is considered in Section 2.4. Section 2.5, presents a review of explicit MPC and implicit MPC along with their strengthens and weaknesses. In Section 2.6, a few MPC algorithms with their the potential to be implemented for fast applications are presented with possible limitations. Finally Section 2.7 concludes the chapter.

2.1 Theoretical foundation of Model Predictive Control

Model Predictive Control (MPC), also referred to as receding horizon control (RHC) or moving horizon or embedded optimisation or predictive control, has been widely adopted in industry as an effective means to deal with multivariable constrained control optimisation problems [3, 5, 6, 9, 27–32]. Several publications provide a good introduction to theoretical and practical issues associated with MPC technology, to name a few: in the area of textbooks there are [3–5, 32–35] as well as the survey papers [6, 9, 28–31, 36–41]. A complete MPC toolbox with different versions have been available since 1998 for Maltlab® [42, 43], showing the interest of the academic community in this field. This

toolbox incorporates different model representations, the ability of constraints handling and the extension to the multivariate case.

The theoretical development of MPC is normally related to the work done around optimal control theory [6]. The relevant literature to the development of MPC deals with the existence of solutions of optimal control problems, characterization of optimal solutions in terms of necessary and sufficient conditions of optimality, Lyapunov stability, and algorithms for the computation of optimal open loop and closed loop feedback controllers [44,45]. The development of optimal control theory concepts can be traced to the work of Kalman in the early 1960's, who determined when a linear control system can be said to be optimal [46, 47]. There are several seminal ideas from optimal control theory literature that are central to MPC. The first two of them are: Hamilton Jacobi Bellman theory (Dynamic Programming) [48], which provides sufficient optimality conditions for determining an optimal feedback controller and the maximum principle [49], which provides necessary optimality conditions to determine an open loop optimal controller given an initial conditions. Other ideas are linear programming which was proposed in the early 1960's by Zadeh and Whalen for solving optimal control problems [50], and the receding horizon concept which was proposed by Propoi [51] in the open loop optimal feedback, who perhaps conceived the first idea of MPC. Later [51], only a few other authors have investigated MPC based on linear programming e.g. [52,53], where the performance index is expressed as the sum of the $\infty-$norm or $1-$norm of the input command and of the deviation of the state from the desired value. Now, there is a comprehensive literature on MPC using linear programming. One motivation is that the resulting optimal control problem is cast as a linear program, which can be computationally less demanding than the corresponding solution of a quadratic program [54, 55].

Solving an infinite horizon, open loop, optimal control problem is computationally expensive (apart from the standard unconstrained case control of linear systems). So it is a relative concern to formulate a receding horizon open loop optimal control problem to solve an infinite horizon problem which provides a stabilising control. The early results were presented in [56] and [57]. Further extensions to these results where provided in [58] and [59]. These extensions show that, implementations of MPC differ from other control methods. Maybe the most important issue for real time implementation is to solve the open loop optimal control problem within a time determined by the sampling instant of the application.

2.2 A brief history of Model Predictive Control

The basic idea of receding horizon control was already indicated by the theoretical work of Propoi [51] in 1963, although it did not gain much attention until the mid 1970's, when Richalet et al. proposed a technique called Model predictive control Heuristic Control and later known as Model Algorithm Control (MAC) [27, 60]. The solution software was referred to as IDCOM, an acronym for Identification and Command. This algorithm employs a finite horizon pulse response (linear) model, a quadratic cost function, and input and output constraints. Engineers at Shell Oil developed their own independent MPC technology in the early 1970's, with an initial application in 1973. Cutler and Ramaker presented details of an unconstrained multivariable control algorithm which they named Dynamic Matrix Control (DMC) at the 1979 National AIChE meeting [61] and at the 1980 Joint Automatic Control Conference [62]. In a companion paper at the 1980 meeting Prett and Gillette [63] described an application of DMC technology in which the algorithm is modified to handle nonlinearities and constraints. The DMC algorithm employs a step response model of the process for the predictions and had huge success in the petro-chemical industry.

These algorithms were heuristic and represent the first generation of predictive controller. Both algorithms provided excellent control of unconstrained multivariable process, but constraint handling, was still somewhat ad hoc. This limitation was overcome in the second generation program, quadratic dynamic matrix control (QDMC) in [64], where quadratic programming is employed to solve the constrained open loop optimal control problem. The third generation of MPC technology, distinguishes between several levels of constraints (soft, hard, ranked), provides some mechanism to recover from an infeasible solution and provides a wider range of process dynamics and controller specifications [9]. Shell multivariable optimising control (SMOC) algorithm is one of the example, which allows for state space models, general disturbance models and state estimation via Kalman filtering [65]. MPC has had a substantial impact on industry that make it a multi million dollar industry with probably exceeding 2000 applications [6].

Industrialist did not address MPC stability theoretically, but achieve stability by restricting attention to stable plants and choosing a large horizon to avoid constraint violations. On the other hand, according to [6], academic research commenced on the theoretical investigation of stability. Initially, stability had to be addressed within the restricted framework of linear analysis because Lyapunov theory was not employed. Research mainly focused on the effect of control and cost horizon on stability when the system is linear, a quadratic cost is used, and hard constraints are absent. Some early approaches proposed by the academic community include Predictive Based Self Tuning Control [66],

Extended Horizon Adaptive Control [67], Multistep Multivariate Adaptive Control [68] and Extended Predictive Self Adaptive Control [69] (i.e., the approaches are not academic, but they were proposed by academics). The Lyapunov function for establishing stability of MPC was first employed in [70], thereafter, the value function was employed as a Lyapunov function for stability analysis of MPC.

Some years later [71, 72] Generalized predictive control (GPC) was presented, which uses some ideas of the generalised minimum variance control [73] and nowadays is one of the most popular predictive control algorithms in academia. This controller uses a CARIMA (Controlled Auto Regressive Integrated Moving Average) model to predict the output of the process and has the characteristics that could be adaptable using a recursive least square parameter estimation. Although at first sight the ideas underlying DMC and GPC are similar, DMC was conceived for multivariable constrained control, while GPC is primarily suited for single variable, and possibly adaptive control [30].

There is another formulation that has had success in the process industry: the Predictive Functional Controller (PFC) developed in [37]. PFC is the product of a company, ADERSA, which had the main aim of maximising the takeup of MPC within industry. It achieved this partially by keeping the algorithm as simple as possible. This algorithm uses a simple optimisation procedure by only taking a subset of points of the control horizon (coincidence points), making for a faster calculation of the control input. Another characteristic of this algorithm is the use of basis functions to structure the control signal that allows the controller to track different set points.

Earlier versions of MPC and GPC did not automatically ensure stability, thus in the 1990's lots of attention was devoted to this topic. There are different proposals and formulations of MPC to ensure stability [6]. One of the standard guidelines to ensure guarantees of feasibility and/or stability are now commonly accepted, that is, many authors use the dual mode prediction paradigm [11, 74] in connection with an infinite horizon. The dual mode prediction is one whereby the predictions have two modes: (i) a transient phase containing degrees of freedom (d.o.f.) and (ii) a terminal mode with guaranteed convergence. The first mode is set up so that a standard prediction mode is assumed, with constraints, up to a particular horizon. Beyond this horizon an unconstrained asymptotically stabilising state feedback control law is assumed with an optimal feedback gain K. However, for a tightly tuned K, regions of attraction (and terminal set) may not be large and the control law may not be robust. In this thesis the standard dual mode prediction set [6, 10, 11, 13] will be adopted as this enables the guarantee of asymptotic stability and recursive feasibility.

MPC is well established and widely used both in industry and control research community

and has reached a high degree of maturity in its linear variant. According to [30, 75] hundreds of papers are reported on predictive control for the years 1991-1998. Currently research is focused on stochastic, nonlinear, large scale, hybrid system and robustness issues as well as fast optimisation or related computational aspects. Nevertheless there are still some theoretical and practical issues which have non-satisfactory answers. For instance, one well understood conflict is how to obtain a large region of attraction, that is the operating region within which the closed loop input, output and state do not violate constraints, and at the same time retain optimum performance. The algorithms that are giving large regions of attraction often give suboptimal performance and vice versa. A simple example of this trade off is the observation that detuning a control law will typically result in smaller input variations so consequently inputs are less aggressive and less likely to violate constraints.

2.3 Nonlinear Model Predictive Control

Nonlinear Model Predictive Control (NMPC) implies controlling a nonlinear plant by means of an MPC control algorithm, where the nonlinearities of the plant are taken into account in some way. Though many manufacturing processes are nonlinear, the majority of MPC applications are based on linear models. This is justified by different reasons. Linear models are usually easy to identify with standard system identification techniques. In addition, MPC have been mostly applied in the oil industry for regulatory control and usually a well identified linear model is detailed enough in order to maintain the process at a desired steady state. Finally, linear MPC converges to an optimal solution in short time, which is the requirement of many manufacturing applications [39, 40, 76].

In [9] several successful applications of fully nonlinear MPC technologies are reported. However, the application of NMPC has not been widely implemented despite a significant amount of research effort having been put into this area. One of the main difficulties facing the transition from MPC to NMPC is the fact that NMPC requires the repeated online solution of a nonlinear optimal control problem. The optimisation problem is generally nonconvex because the model equations are nonlinear, and therefore computationally expensive [76]. The problem of the existence of an online solution of the nonlinear program is crucial one. This is one of the key limiting factors for successful practical applications of NMPC.

2.4 Robust Model Predictive Control

The success of MPC depends on the degree of precision of the plant model. In practice, modelling real plants inherently include uncertainties that have to be considered in the controller design, that is a control design procedure has to guarantee robustness properties such as stability and performance of the closed loop system in the whole uncertainty domain. Two typical descriptions of uncertainty, state space polytope and bounded unstructured uncertainty are extensively considered in the field of robust model predictive control.

Two common approaches to robustness have been considered in the MPC literature. The first aspect is the inherent robustness of nominal MPC algorithm design, i.e. on the MPC algorithm that is not specifically designed for robust aspects (like stability and performance) [77, 78]. The second aspect is the explicit inclusion of robustness requirements into the design of an MPC algorithm has received attention in the MPC literature. MPC algorithm design based on including the uncertainty information in the model is referred to as robust MPC [21, 79–83].

Traditionally robust MPC requires the solution of a min-max optimisation problem, where an optimisation over all possible control moves is performed in order to minimise a worst-case (over all possible uncertainty realisations) cost function [79, 84–86]. Furthermore, constraint satisfaction also has to be guaranteed for all possible future trajectories. In general, solving a min-max problem subject to constraints, one has to optimise over a sequence of control strategies rather than a sequence of fixed control moves, all these elements contribute to make robust MPC often intractable for online optimisation [87]. However, most robust MPC can be classified into open-loop min-max MPC [84,88,89] and feedback min-max MPC [21, 79, 80, 87, 90, 91]. An open-loop model formulation usually overestimates the uncertainty in the closed-loop process because it does not consider the effects of feedback that will occur in the future to reduce the effect of model mismatch. Therefore, most of the proposed robust MPC algorithms optimise the worst case cost function over a sequence of feedback control laws. This improvement over the open-loop MPC approach is achieved at a considerable increase in computational complexity. The robust stability of the closed-loop min-max MPC is achieved either directly by enforcing a type of robust contraction constraint such as a robustly invariant sets [92, 93], or by minimising the case worst performance over a specified uncertainty range; the corresponding min-max performance optimisations lead to robust stability [30, 38].

The contractive MPC was first introduced in [94] and a stability proof was developed by the authors in [95]. The main idea in this approach is to add contractive constraints to the usual formulation of the MPC which enforces the actual state to contract at discrete

intervals in the future. The system behaviour will converge as long as there exist solutions which satisfy the contraction constraint. Stability can be established using a Lyapunov function [38]. For robust MPC, the necessary and sufficient conditions for robust stability are proposed in [92] using contraction properties of MPC. The paper [96] showed how to determine the weights such that robust stability can be guaranteed, however the proposed technique can be very conservative because it assumed an independent uncertainty bound on the FIR model coefficients.

Furthermore a dynamic programming problem is formulated in [90] to minimise the worst case cost but the approach suffers from a large computational burden and non-convexity [38]. Consequently this approach is intractable except the low order systems with simple uncertainty descriptions. Other formulations [90,97] approximated the problem by simplifying the objective and the uncertainty description to reduce the online computational burden [38]. All these approaches are conservative for certain problems. In [93], the authors propose the use of nominal performance and robust stability is achieved by enforcing robust contraction constraints. Another suggestion with a cost contraction constraint is proposed in [98] to formulate a convex optimisation problem.

However, various alternative robust MPC algorithms have been proposed to approximate solutions of the max-min problem but with a reduced computational burden. In [79] a classical result is presented by directly incorporating the plant uncertainty into the MPC formulation. The existence of a feedback law minimising an upper bound on the infinite horizon objective function and satisfying the constraints is reduced to a convex optimisation using linear matrix inequalities (LMI). The main disadvantages are that the use of LMI-based optimisation can be computationally demanding because the optimisation problem needs to be solved on-line at each sampling instant and contains many decision variables and constraints. Moreover the methods use conservative constraint handling. Since, the algorithm is derived by using a single Lyapunov function, the algorithm turns out to be very conservative.

To simplify the computational complexity, many robust MPC algorithms approximate the future controller behaviour with affine feedback control laws [21,79,99–102]. By using control invariant sets and LMIs, a broad class of model uncertainty can be addressed with guaranteed robust closed-loop stability. In order to improve the control performance and decreased the computational burden at the same time, many synthesis approaches of MPC were proposed in [103–106]. Based on the concept of the invariant set several papers [10, 87, 107, 108] separated the computational load into off-line (construct a sequence of explicit control laws corresponding to a sequence of invariance sets) and online (calculate the control input with low computational burden). Recently the authors of [109–111] have proposed a systematic way to derive a sequence of state feedback control laws to

enlarge the size of the region of attractions.

Another suggestion which is little considered in the literature is the concept of triple mode control [112, 113]. In this strategy one recognises that a large region of attraction in conjunction with good performance often implies a nonlinear or linear time varying (LTV) prediction dynamics. The challenge is to find a suitable and LTV control law which enlarges the region of attraction without too much detriment to performance. In [114], the authors proposed robust triple mode MPC using prediction dynamics in [25] to enlarge the region of attraction for robust MPC. In [25], the authors proposed a convex formulation of prediction dynamics and showed that the resulting maximal ellipsoidal set is equal to the maximal invariant ellipsoidal set under any linear feedback law. They employed a nonlinear transformation of variable to allow for polytopic uncertainty in the model parameters and linear input/state constraints. In robust triple mode MPC, the computational burden is further simplified using the reduced-complexity invariant sets in [115] for the case of quasi-infinite horizon closed loop MPC. The reduced-complexity invariant sets may result in a decrease in the number of on-line optimisation variables [115]. This invariant set structure is used in the design of robust MPC and this thesis will pursue this type of approach to including uncertainty information in the model.

2.5 Real time implementation

MPC approaches determine a sequence of optimal control (input) over a future time horizon in order to optimise the performance of the controlled system, expressed in terms of a cost function. The optimisation is based on an internal mathematical model which, given the current measurements, predicts the future behaviour of the real system with respect to changes in the control inputs. Once the sequence of optimal control inputs has been determined, only the first element is actually implemented and the optimisation is repeated at the next time interval with the new measurements and over the shifted horizon. This feedback mechanism of MPC compensates for the prediction error due to structural mismatch between the internal model and the real system as well as for disturbances.

One important issue for real time implementation is to solve an optimisation problem within the time determined by the sampling instant of the application and therefore the computational efficiency of an algorithm becomes critical. A trade off has to be made between performance, region of attraction and the computational burden when choosing from the currently available algorithms; in simple terms a larger region of attraction usually implies a higher computational load or worse performance. It is also recognised

that the range of industrial applications of MPC has been restricted in practice due to the computationally expensive online optimisation which is required. There are essentially two popular ways to solve the finite horizon optimal control problem, namely explicit and implicit.

2.5.1 Explicit MPC

Explicit MPC formulations [116–121], based on multiparametric programming [122,123], move much of the optimisation effort offline and obtain the optimal control as an explicitly defined piecewise affine (PWA) function with dependence on the current state vector. The domain of the PWA function is the set, which is partitioned into convex regions. This allows the online computational effort to be reduced to a series of function evaluations, eliminating the need of a real-time optimisation solver. Therefore, explicit MPC represents a promising approach to extend the scope of applicability of MPC to situations where the computations required for the online optimisation are restrictive, and/or where insight into the control behaviour is necessary for performance analysis (like safety verification). These situations are common, for example, in the medical, automotive and aerospace industries [124, 125]. MPC functionality can, with this, be applied to applications with sampling times in the milliseconds range, using low cost embedded hardware [126]. Software complexity and reliability is also improved, allowing the approach to be used in safety-critical applications.

However, an explicit MPC implementation may still be prohibitively costly for large optimisation problem. In fact the major drawback with these approaches is that the storage requirements, and online search, may grow exponentially with the optimisation variables, states and input dimensions, so that the 'explicit solution' is often only actually efficient for small problems (where dimension is no more than around 5) [127]. Several authors have proposed various research directions to solve this problem. These proposals are mainly based on post processing the feasible set partitions to reduce the complexity of the PWA function. Some authors [128] have accepted that there could be no reduction in the number of regions, but there may be more efficient ways to identify the active regions. In [129, 130] authors proposed the explicit PWA solution is post processed to generate search trees that allow efficient online evaluation. In [131], the feasible set partition regions were merged with the same feedback law and thereby reducing the complexity of the PWA function. A few authors in [132–134] have considered a suboptimal parametric solution in the hope that such a solution may be far simpler, but with a small loss in performance. They defined regions as hypercubes as this allowed a very efficient search. An even less explored avenue is the potential to use interpolation [16] to give a convex

blend from nearby points, thus reducing the number of point/regions required while ensuring feasibility. Another less explored avenue is to base the parametric solution on points rather than regions [135,136]. A different approach was discussed in [137] and [138] to enlarge the region of attraction with only a small detriment to optimality but in such a way that the resulting predictions class can be combined with parametric programming to give far simpler solutions.

The possible advantages deriving from explicit MPC approaches have attracted a lot of interest in the research community and considerable effort has been put into the development of techniques to deal with the entailed disadvantages.

2.5.2 Implicit MPC

Until the 90s MPC was only used for plants with slow dynamics. It was widely applied in petrochemical and related industries where satisfaction of constraints was particularly important because efficiency demands operating points on or close to the boundary of the set of admissible states and control. One of the primary advantages of this technique is its explicit capability to handle constraints. However, the fact that the optimisation procedure is to be repeated every time step, is the reason that the application of the MPC has been limited to the slow dynamics of systems in the process industry until recently. The boom in MPC started in 1990s when faster computers became available together with the rapid development of optimisation algorithms. These days MPC applies to various types of plants with fast dynamics such as airplanes, satellites, robotics, automotive etc.

In MPC, the control action at each step is computed by solving an online optimisation problem. The optimisation problem resulting due to the linear model, polyhedral constraints, and a quadratic objective is a quadratic program (QP). Solving the QP using standard techniques can be slow, and this has traditionally limited MPC to applications with slow dynamics, with sampling time measured in seconds or minutes. The computational efficiency of an algorithm in this aspect becomes critical. There is a collection of methods that can be used to speed up the computation of MPC control law, using online optimisation.

Two approaches to solving the QP problem are generally used: Active Set Methods (ASMs) [139–141] and Interior Point Methods (IPMs) [127, 142–145]. The historical perspective of the computational complexity of ASMs and IPMs is described in [146]. The computational complexity of the ASM is exponential in the worst case [147], while in practice they are efficient and the number of iterations required is typically a small polynomial function of the dimensions [143]. Average-case computational complexity

analysis has shown the discrepancy between practice and the worst-case behaviour of ASMs [148]. For IPMs, the theoretical bound is polynomial in the sizes of the matrices in the QP, and in practice the number of iterations these methods perform is essentially independent of the number of system dimension and number of constraints [149–151]. They perform large amount of work per iteration, the ratio of work to communication is quite large compared to ASM. Thus, they are able to exploit the advance computing architectures, as well as sparse linear solvers and including parallel methods to solve linear systems. ASMs, which perform a small amount of work per iterations but the ratio of work to communication is small [151, 152].

The ASM gives a systematic means of selecting a potential active set and iterating through these potential sets to find the global optimum. It generates a sequence of approximate solutions and maintains and updates a prediction of the optimal sets of active and inactive constraints. A warm start strategy [143, 153] can be used to exploit an advanced starting point for solving the QP and thus to further improve the computational efficiency. This strategy may significantly reduce the number of iterations required and thus improve the computational effort. ASMs are generally more efficient than IPMs for small or medium scale problems involving few constraints [152]. In [154], a sampling time of the order of millisecond is achieved in MPC using online active set strategy for real world diesel engine problems.

IPM is becoming more popular within MPC implementation because they are suited to large, sparse problems where the linear algebra can be done fast. However, the associated optimisation at each iteration is more demanding. In [155], IPM has proven to be an efficient way of solving linear, quadratic, and nonlinear programming problems compared to ASMs especially for large scale problems. However, they cannot exploit the solution of the preceding problem using warm start active set methods proposed in [143, 156]. A warm start of an IPM with the solution of the preceding problem if it is close to the boundary of the region of attraction, usually leads to the blocking the search direction, which means that the step length becomes very small, and the next iteration will be very close to the previous one. In the last decade, a number of attempts have been made to improve warm start strategies in IPMs. A warm start strategy for Linear programming (LP) problem was discussed in [157, 158], where a worst case estimate on the number of a perturbed LP is determined. Their estimate mainly depends on the size of the perturbation and on the conditioning of the original problem. It is concluded in [158] that most of the strategies are effective in reducing the computational time for small perturbations. Recently, a new unblocking strategy was proposed by [156]. This is based on a sensitivity analysis of the search direction with respect to the current point. Numerical test shows that, on average the computations can be simplified in a

range of LP and QP problems varying from small scale to large scale problems when this unblocking strategy is combined with other warm-start strategies. In [159], warm-starting and early termination of the QP problem is proposed.The early termination significantly reduces the computations, but on the other hand it may compromise either guaranteed feasibility or stability under specified fixed runtime for the general case of state and input constraints [160]. An extension to design warm-start strategy was also proposed in [160–163].

In recent years, attempts have been made to use predictive control in fast processes with a short sampling time. To reduce the computational efforts new techniques have emerged. Even though modern advances in computing technology allow faster sampling rates, a critical that issue remains concerns the reliability and verifiability of the control algorithm. Moreover, the implementations via real time solvers are not well suited for all the situations which require portable and/or embedded control devices. Depending on the particular problem properties and implementation restrictions, the user then has to decide for one of the two approaches (explicit or implicit MPC). Authors in [163,164], proposed different strategies to overcome an existing gap of problem sizes and types, which are either intractable for explicit MPC or implicit MPC solution to meet the required online computation times.

2.6 Efficient MPC algorithms

The development of dual mode paradigms/infinite horizon algorithms solved a theoretical problem for the academics and hence gave MPC some analytical rigor. The insight gained is invaluable in understanding better the strengths and weaknesses of more typical MPC implementations. All algorithms to some extent allow a trade off between performance, MCAS volume and the computational burden when choosing from the currently available algorithms. It is recognized that the range of industrial applications of MPC has been restricted in practice due to the computationally expensive online optimisation which is required.

Several authors have looked at this problem, although less so in recent years, where the focus has moved more to nonlinear systems, robust and parametric solutions. This section presents a review of a selection of existing proposals along with their strengths and weaknesses.

2.6.1 Optimal MPC (OMPC)

The standard dual mode prediction set up [10, 11] has two tuning parameters. The most obvious is the number of degrees of freedom (d.o.f. or n_c) but the second and in practice equally important is the selection of the terminal control gain (i.e. K). One logical choice [10, 11, 165] for K is in fact that which minimises the infinite horizon cost in the constraint free case. Define such an algorithm as linear optimal MPC (OMPC). The advantage of such a choice is that for any n_c, if the unconstrained optimum is feasible, then the dual mode algorithm will find that solution; that is no prediction mismatch in the unconstrained case. Moreover, if n_c is large enough, then the dual mode algorithm can find the optimal for the constrained infinite dimensional optimisation. In practice, for computational (and sometimes robustness) reasons, this n_c may not be very large [5].

However, the major weakness is that for a well-tuned K to give high performance often have relatively small regions of attraction (regions where the class of predictions satisfy constraints) unless one uses a prohibitively large number of d.o.f. (or n_c).

2.6.2 Interpolation

Interpolation methods have looked for alternative ways to formulate the d.o.f. for optimisation [14, 15, 166, 167]. These methods are based on two control laws; these are the desired (highly tuned) control law and a detuned control law. It seeks to combine the attributes of control strategies with known properties, for instance, one with enlarging the region of attraction and one with improved performance, while utilising a small number of d.o.f.; however, early stability and convergence results [15, 16] are weak. A later suggestion, GIMPC or generalised interpolation MPC [166], was focused on the uncertain cases and thus restricted to ellipsoidal regions, although later extended to polyhedral regions [168, 169]. GIMPC and GIMPC2 [170], includes sufficient and necessary details to remove the conservatism of early proposed strategies.

However, interpolation methods do not currently extend well to large dimensional systems and, as they do not fit as conveniently into a normal paradigm, more work is required to encourage take up by colleagues and industry [171].

2.6.3 Move blocking MPC

Some authors [20, 172, 173] have proposed blocking as means to deal with the computational burden of optimal control. It is common practice to reduce the d.o.f. by fixing the

input or its derivatives to be constant over several time steps. This policy is referred to as 'move blocking'.

However, the feasibility and convergence analysis of such algorithms are not straightforward due to the prediction dynamics. This is because the prediction dynamics do not automatically retain the prediction tail (i.e. Bellman's principle of optimality - any segment of optimal trajectory is also optimal [48]). In order to retain recursive feasibility guarantees, the blocking strategy must be time-varying [17,174], or additional constraints need to be enforced on the first prediction step [20,175].

2.6.4 Triple mode MPC

Some interesting work considered the so-called triple mode MPC [25,112–114,176], where one embeds a smooth transition between a controller with enlarging the region of attraction and another with improved performance into a single mode model and uses the decision variables to improve performance/region of attraction further. This strategy often implies nonlinear or time varying (LTV) prediction dynamics [177]. Hence, a sensible objective is to find a suitable and fixed LTV control law which enlarges the region of attraction without too much detriment to performance.

The further study needs to make the algorithm handle non-symmetric constraints better, and to develop triple mode algorithms for uncertainty classes with bounded state disturbances.

2.6.5 Laguerre Optimal MPC

A general prediction framework was proposed in [25] to enlarge the region of attraction. Laguerre functions were proposed as a special case of this framework deploying a particular lower triangular structure in [26]. The simpler structure can be explored to come up with an optimisation of the volume of polytopic controller sets. This special structure also allowed only one tuning parameter to generate the prediction framework.

Laguerre functions were first used in the context of a GPC type of algorithm [2,18], but more recently it was noted that they provide a possible solution to the volume/complexity trade off in OMPC [26]. Conceptually the key point is to parameterise the future values of input perturbations in terms of Laguerre functions and this means that the values do not go to zero after n_c steps, but rather converge asymptotically. Therefore, this removes the need for the state prediction to enter the terminal control region (i.e. maximal admissible

set MAS [12]) in n_c steps because the input parameterisation does not revert straight back to the unconstrained optimal, but rather approaches the unconstrained optimal gradually as the perturbation terms reduce in size.

Nevertheless, it needs further study to identify systematic ways of choosing the best 'Laguerre function parameterisations or indeed any alternative choices to improve the trade off within an OMPC formulation.

2.7 Conclusion

In this chapter, the literature overview of model predictive control beginning from its theoretical foundations in optimal control theory and the early model predictive control approaches in an historical context were presented. This was followed by a brief discussion about nonlinear model predictive control strategies with implementation limitations. One of the important aspects of real time implementation is the robustness of the design algorithm in presence of model uncertainties. A brief overview of already proposed robust design schemes was discussed. For real time implementation, there are essentially two popular ways to implement the model predictive control algorithms, an explicit and an implicit solution of the finite horizon optimal control problem. A review was conducted for both implementation solutions. It has been noted that there is a well understood trade off between the region of attraction, performance and inexpensive optimisation implementation of model predictive control algorithms. Finally, a quick overview of efficient algorithms was given, which was already in the mainstream journals to simplify this trade off along with their strengths and weaknesses.

The next two parts of this thesis will develop solutions to enlarge the region of attraction and improve performance further using low computational complexity model predictive control algorithms. The efficient algorithms including Optimal MPC, Laguerre optimal MPC and triple mode MPC will be considered as underlying algorithms with recursive feasibility and asymptotic stability in nominal cases. Both linear and robust analyses will done in part II and part III respectively. The next chapter will discuss in detail the mathematical formulation of the model predictive control problem, dual mode model predictive control, optimal MPC, Laguerre optimal MPC and triple mode MPC for both linear and robust scenarios.

Chapter 3

Overview of Model Predictive Control Algorithms

Model predictive control (MPC) is an advanced technique that has had a great impact on industrial control engineering. This chapter overviews some standard MPC algorithms with emphasis on efficient algorithms in order to identify the gaps within the literature. It provides the mathematical formulation needed for the successive chapters. Most of the definitions and results are well established and can be found in the literature. Other definitions are slightly adapted in the framework of this thesis.

The chapter starts with the problem formulation of standard predictive control and stability in MPC is presented in Section 3.1. A dual mode prediction paradigm with an optimal MPC (OMPC) is presented in Section 3.2. Section 3.3 presents the Laguerre functions and Laguerre optimal MPC (LOMPC) algorithm. The triple mode prediction paradigm is presented in Section 3.4. Section 3.5 presents a robust MPC problem formulation using linear parameter varying system with constraints. A robust dual mode algorithms using polyhedral and ellipsoidal invariant sets are presented in Section 3.6. It is further extended in Section 3.7 to consider triple mode prediction paradigm. Section 3.8 summaries the key points and discusses the limitation in already proposed algorithms. Finally, Section 3.9 gives the conclusion of the chapter.

3.1 Problem formulation for Predictive Control

This section introduces the assumptions used in this thesis and background information, but omitting the well-known algebra that does not add to concepts.

3.1.1 Model and constraints

This thesis consider a discrete time, controllable and observable, linear time invariant state space model of the form

$$x_{k+1} = Ax_k + Bu_k, \qquad k \in \mathbb{N} \tag{3.1}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector and $u \in \mathbb{R}^{n_u}$ is the control input, $A \in \mathbb{R}^{n_x \times n_x}$ and $B \in \mathbb{R}^{n_x \times n_u}$ are matrices defining the actual behaviour of the system. Model uncertainty can be added to this formulation (e.g. [115]) but is omitted in this section to ensure the presentation of key steps and concepts are as clear and straightforward as possible [1].

Assume that the states and inputs at all time instants should fulfill the following constraints.

$$\begin{aligned} x_k &\in \mathbb{X} \subset \mathbb{R}^{n_x} \\ u_k &\in \mathbb{U} \subset \mathbb{R}^{n_u} \end{aligned} \tag{3.2}$$

The sets $\mathbb{X}$ and $\mathbb{U}$ are considered to be described by linear inequalities on the respective variables.

$$\begin{aligned} \mathbb{X} &= \{x_k \in \mathbb{R}^{n_x} \mid \underline{x} \leq x_k \leq \overline{x}\} \\ \mathbb{U} &= \{u_k \in \mathbb{R}^{n_u} \mid \underline{u} \leq u_k \leq \overline{u}\} \end{aligned} \tag{3.3}$$

In further sections it will assume that $\mathbb{U}, \mathbb{X}$ are convex, compact sets containing the origin in their interior.

[1] Full state measurement and no disturbance are assumed.

3.1.2 Cost function

In the context of predictive control, it is common to take the following quadratic cost function as the objective to be minimised at each sample

$$J(x_k, u_k) = \sum_{i=0}^{\infty} \{x_{k+i+1}^T Q x_{k+i+1} + u_{k+i}^T R u_{k+i}\}, \tag{3.4}$$

where $Q = Q^T \succeq 0$ and $R = R^T \succ 0$ are the state and input cost weighting matrices respectively. When no constraints are considered, the infinite horizon objective function (3.4) is minimised by the time-invariant feedback [178]

$$u_k = -Kx_k. \tag{3.5}$$

When constraints on the state, input and outputs (3.2) are considered, an analytical form of the optimal control law such as (3.5) does not exist. Therefore, to achieve feedback the (open loop) minimisation of the cost function needs to be performed at each sampling instant when x_k is available, and apply the corresponding u_k, that is the first part of the optimal input sequence. The main issue here is that the resulting optimisation problem is generally intractable due to the infinite number of optimisation variables.

Remark 3.1.1. This thesis avoids the detail associated to non zero set points and disturbances [179, 180] which can be incorporated by using the feedback law $[u_k - u_{ss}] = -K(x_k - x_{ss})$ where u_{ss}, x_{ss} are the states expected to give offset free tracking. Incorporating integral action into a state feedback is equivalent to finding consistent estimates of the steady-state values of the state and input; that is, x_{ss}, u_{ss} using disturbance estimate. A good discussion of this can be found in [179].

However, it has been shown [181, 182] that it is possible only for n_c large enough to optimise the cost function over the infinite horizon with a finite number of optimisation variables if the cost function is viewed as composed by two parts

$$\begin{aligned}\sum_{i=0}^{\infty} \{x_{k+i+1}^T Q x_{k+i+1} + u_{k+i}^T R u_{k+i}\} = &\sum_{i=0}^{n_c-1} \{x_{k+i+1}^T Q x_{k+i+1} + u_{k+i}^T R u_{k+i}\} \\ &+ \sum_{i=n_c}^{\infty} \{x_{k+i+1}^T Q x_{k+i+1} + u_{k+i}^T R u_{k+i}\}\end{aligned} \tag{3.6}$$

where $n_c < \infty$ corresponds to a chosen horizon. It is noted that after some time the

constraints are satisfied naturally [2], and assuming n_c horizon, the control inputs in the first part are the only optimisation variables that need to be considered, since the control inputs in the second part are given by linear quadratic regulator (LQR). Therefore second part is given by

$$\sum_{i=n_c}^{\infty} \{x_{k+i+1}^T Q x_{k+i+1} + u_{k+i}^T R u_{k+i}\} = x_{n_c+k} Q_f x_{n_c+k} \tag{3.7}$$

where $x_{n_c+k} Q_f x_{n_c+k}$ is known as the terminal cost function and Q_f is determined from Lyapounov equation [5].

$$(A - BK)^T Q_f (A - BK) = Q_f - (A - BK)^T Q (A - BK) - K^T R K \tag{3.8}$$

The presence of the constraints in the optimisation problem results in the concepts of feasible set [142].

Definition 3.1 (Feasible set)**.** *The feasible set $\mathcal{X}$ is defined as the initial states x_0 for which constrained optimisation of cost function in* (3.6) *subject to* (3.3) *and* (3.1) *with n_c horizon is feasible i.e.*

$$\mathcal{X} = \{x_0 \mid \exists \{u_0, \ldots, u_{n_c-1}\} \textit{ satisfying } (3.3) \textit{ and } (3.1)\}. \tag{3.9}$$

An optimal input satisfying the constraints is guaranteed to exist for any initial state inside the feasible set.

One of the advantages of having an infinite horizon is that if the initial set is chosen inside the feasible set, the nominal closed loop (exponential) asymptotic stability is ensured [6]. As infinite horizon cost function can be written as (3.6) which allows the formulation of a tractable optimisation problem in the presence of constraints. It guarantees the asymptotic stability as long as constraints will not be violated after the end of the chosen horizon n_c.

One possibility to achieve feasible solution after n_c is to select a long enough horizon to guarantee that constraints will be satisfied afterwards [182]. However, this may result in an expensive optimisation to solve the corresponding open loop optimisation problem. A

[2] It is assumed that the state meets the constraints.

preferred approach is to introduce a terminal state constraint

$$x_{n_c} \in \mathcal{X}_0 \tag{3.10}$$

where $\mathcal{X}_0$ is called the terminal set. With the introduction of terminal set, asymptotic stability can be guaranteed for any horizon length. After introducing the terminal set the length of horizon n_c sets the number of degrees of freedom in the optimisation formulation. The feasible set is then directly dependent upon n_c.

3.1.3 Problem formulation

A generic MPC problem can be formulated using (3.1), (3.3) and (3.6) as:

$$\min_{u_0,\ldots,u_{n_c-1}} \{J(x_k, u_k) = x_{n_c+k} Q_f x_{n_c+k} + \sum_{i=0}^{n_c-1} \{x_{k+i+1}^T Q x_{k+i+1} + u_{k+i}^T R u_{k+i}\}\} \tag{3.11}$$

Subject to

$$\begin{aligned} &x_{k+1+i} = Ax_{k+i} + Bu_k \\ &x_{k+i} \in \mathbb{X} \subset \mathbb{R}^{n_x} \\ &u_{k+i} \in \mathbb{U} \subset \mathbb{R}^{n_u} \\ &x_{n_c} \in \mathcal{X}_0 \\ &i = 0, \ldots, n_c - 1. \end{aligned}$$

3.1.4 Stability in MPC

The MPC optimisation problem formulation (3.11) in the presence of constraints makes the closed loop system nonlinear and hence, the stability analysis is more complex; typically the use of Lyapunov stability theory [183] is effective. The main idea is to modify the MPC concept such that the cost function can be used as Lyapunov function to establish stability [6]. The terminal cost and the terminal constraints are introduced to explicitly ensure stability and recursive feasibility [4–6, 11, 70, 181, 182, 184]. The stability proof can be found in [6] using terminal cost and terminal constraints set using [12].

3.2 Dual mode MPC

The dual mode terminology was originally proposed by Sznaier in [184] but now most authors accept the usefulness of a dual mode paradigm for guaranteeing nominal stability for MPC as done in previous section [4–6,11,181,182]. Consider MPC predictions to have two modes [5]

1. A transient phase containing degree of freedom (d.o.f.); typically in this case of prediction the state could be some distance from steady state.
2. A terminal mode with guaranteed convergence. It is normal for mode 2 behaviour to be given by a known control law (3.5).

In terminal mode, it is now common practice to make use of positive invariant sets in order to establish recursive feasibility of the proposed optimisation and the region of attraction within which the chosen algorithm can operate reliably. The subsequent section establishes the concept of the maximum admissible sets (MAS) as the largest possible positive invariant set to be used as a terminal set.

Definition 3.2 (Positive invariant set)**.** *Given a dynamical system $x_{k+1} = f(x_k)$ and x_0 is the initial point. The set $\mathcal{X}_0$ is said to be positively invariant if*

$$x_0 \in \mathcal{X}_0 \implies x_k \in \mathcal{X}_0, \qquad k = 1, 2, \ldots \tag{3.12}$$

Intuitively, this means that once a trajectory of the system enters $\mathcal{X}_0$, it will never leave it again [185].

3.2.1 Maximum Admissible Set (MAS)

Assume a stabilising linear state feedback gain K, which is applied as a control law to regulate (3.1) to the origin and yielding closed loop state transition matrix $\Phi = A - BK$. Then the MAS can be defined as [12]

Definition 3.3 (Maximum Admissible Set (MAS))**.** *Let $\mathcal{X}_0$ (maximal admissible set MAS) be the polytopic control invariant set for which all constraints are satisfied [12]*

$$\mathcal{X}_0 = \{x \in \mathbb{R}^{n_x} \mid \Phi^i x \in \mathbb{X}, \ -K\Phi^i x \in \mathbb{U}, \forall i \geq 0\}. \tag{3.13}$$

Although consideration of constraints on an infinite horizon appears to be intractable for stable systems where steady state does not lie on constrained boundary there will exist a finite horizon such that the further future constraints are redundant [12]. An algorithm for computation of the MAS assuming Lyapunov stability can be found in [12].

3.2.2 Optimal MPC (OMPC)

There are two main tuning parameters affecting dual mode control. The most obvious is the number of d.o.f. but the second and in practice equally important one is the selection of the terminal control law (3.5). The choice of terminal control law has a significant impact on unconstrained and constrained performance for a small number of d.o.f.. One logical choice [5, 10, 11, 181] for terminal control gain K is in fact that which minimises the infinite horizon cost in constraint free case. Define such an algorithm as an optimal MPC (OMPC).

The input predictions are defined as follows

$$u_{k+i} = \begin{cases} -Kx_{k+i} + c_{k+i} & \forall i = 0, \ldots, n_c - 1 \quad \text{Mode 1} \\ -Kx_{k+i} & \forall i \geq n_c \quad \text{Mode 2} \end{cases} \tag{3.14}$$

where the perturbations c_k are the d.o.f. (or control moves) for optimisation; conveniently summarised in vector form as $\underset{\rightarrow}{c}_k = [c_k^T, \ldots, c_{k+n_c-1}^T]$. In essence $\underset{\rightarrow}{c}_k$ are the perturbations about the unconstrained optimal input trajectory required to meet constraints.

3.2.3 Closed loop prediction implementation of dual mode OMPC

For completeness this section gives the closed loop predictions which can be deployed in MPC algorithms [5]. The state space form using the prestabilised loop (during predictions) with $x_{k|k} = x_k$ are

$$x_{k+i|k} = Ax_{k+i-1|k} + Bu_{k+i}; \quad u_{k+i} = -Kx_{k+i|k} + c_{k+i} \tag{3.15}$$

Removing the dependent variable u_{k+i} one gets

$$x_{k+i|k} = [A - BK]x_{k+i-1|k} + Bc_{k+i}; \quad u_{k+i} = -Kx_{k+i|k} + c_{k+i} \tag{3.16}$$

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

$$\underset{\rightarrow}{x}_k = \underbrace{\begin{bmatrix} \Phi \\ \Phi^2 \\ \Phi^3 \\ \dots \end{bmatrix}}_{P_{cl}} x_k + \underbrace{\begin{bmatrix} B & 0 & 0 & \dots \\ \Phi B & B & 0 & \dots \\ \Phi^2 B & \Phi B & B \dots & \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{H_c} \underset{\rightarrow}{c}_k \tag{3.17}$$

$$\underset{\rightarrow}{u}_k = \underbrace{\begin{bmatrix} -K \\ -K\Phi \\ -K\Phi^2 \\ \dots \end{bmatrix}}_{P_{clu}} x_k + \underbrace{\begin{bmatrix} I & 0 & 0 & \dots \\ -KB & I & 0 & \dots \\ -K\Phi B & -KB & I \dots & \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{H_{cu}} \underset{\rightarrow}{c}_k \tag{3.18}$$

or in compact form

$$\underset{\rightarrow}{x}_k = P_{cl} x_k + H_c \underset{\rightarrow}{c}_k \tag{3.19}$$

$$\underset{\rightarrow}{u}_k = P_{clu} x_k + H_{cu} \underset{\rightarrow}{c}_k \tag{3.20}$$

Similarly including constraints

$$\begin{bmatrix} \underline{x} \\ \underline{x} \\ \vdots \end{bmatrix} \leq \underset{\rightarrow}{x}_k \leq \begin{bmatrix} \overline{x} \\ \overline{x} \\ \vdots \end{bmatrix}; \quad \begin{bmatrix} \underline{u} \\ \underline{u} \\ \vdots \end{bmatrix} \leq \underset{\rightarrow}{u}_k \leq \begin{bmatrix} \overline{u} \\ \overline{u} \\ \vdots \end{bmatrix}; \tag{3.21}$$

Finally, from prediction class (3.19), (3.20) and (3.21) can be presented as

$$M_{cl} x_k + N_{cl} \underset{\rightarrow}{c}_k \leq d_{cl} \tag{3.22}$$

for suitable M_{cl}, N_{cl} and d_{cl} prediction matrices with finite number of rows [5, 12]. Two standard sets can be define as

Definition 3.4 (Maximum Control Admissible Set MCAS)**.** *Let $\mathcal{X}_c$ (maximal control admissible set MCAS) be the set of initial states x_k for which the prediction parameterisation (3.14) is feasible (that is satisfies constraints), then*

$$\mathcal{X}_c = \{x_k \in \mathbb{R}^{n_x} \mid \exists \ \underset{\rightarrow}{\tilde{c}}_k \in \mathbb{R}^{n_c n_u}, Mx_k + N \underset{\rightarrow}{\tilde{c}}_k \leq d\} \tag{3.23}$$

and MAS can be define in compact form as

$$\mathcal{X}_0 = \{x_k \in \mathbb{R}^{n_x} \mid M_0 x_k \leq d_0\}. \tag{3.24}$$

Remark 3.2.1. The open loop paradigm and closed loop paradigms give an identical prediction class [5]. The closed-loop paradigm uses perturbations of the unconstrained optimal control law as d.o.f.. This gives good insight into the impact of constraints on performance [5].

3.2.4 Cost function using closed loop paradigm

The derivation of the cost function is similar to that provided in Lemma 4 in [186]. Consider the closed loop state predictions using (3.1) and (3.14)

$$x_{k+i} = \begin{cases} \Phi x_{k+i} + B c_{k+i} & \forall i = 0, \ldots, n_c - 1 \\ \Phi x_{k+i} & \forall i \geq n_c \end{cases} \tag{3.25}$$

Substituting definitions (3.14) and (3.25) into (3.11)

$$\begin{aligned}
J_k &= \left\|x_{k+n_c}\right\|_{Q_f}^2 + \sum_{i=0}^{n_c-1} \left[\left\|x_{k+1+i}\right\|_Q^2 + \left\|-Kx_{k+i} + c_{k+i}\right\|_R^2\right] \\
&= \left\|\Phi x_{k+n_c-1} + B c_{k+n_c-1}\right\|_{Q_f}^2 + \left\|x_{k+n_c}\right\|_Q^2 + \left\|-Kx_{k+n_c-1} + c_{k+n_c-1}\right\|_R^2 \\
&+ \sum_{i=0}^{n_c-2} \left[\left\|x_{k+1+i}\right\|_Q^2 + \left\|-Kx_{k+i} + c_{k+i}\right\|_R^2\right] \\
&= \left\|x_{k+n_c-1}\right\|_{\Phi^T Q_f \Phi + K^T R K + Q}^2 + \left\|c_{k+n_c-1}\right\|_{B^T Q_f B + R}^2 \\
&\quad + 2x_{k+n_c-1}(\Phi^T Q_f B - K^T R) c_{k+n_c-1} \\
&\quad + \sum_{i=0}^{n_c-2} \left[\left\|x_{k+1+i}\right\|_Q^2 + \left\|-Kx_{k+i} + c_{k+i}\right\|_R^2\right] \\
&= \left\|x_{k+n_c-1}\right\|_{Q_f}^2 + \left\|c_{k+n_c-1}\right\|_{B^T Q_f B + R}^2 \\
&\quad + 2x_{k+n_c-1}(\Phi^T Q_f B - K^T R) c_{k+n_c-1} \\
&\quad + \sum_{i=0}^{n_c-2} \left[\left\|x_{k+1+i}\right\|_Q^2 + \left\|-Kx_{k+i} + c_{k+i}\right\|_R^2\right]
\end{aligned}$$

where $Q_f = \Phi^T Q_f \Phi + K^T R K + Q$. Note that

$$\begin{aligned}
\Phi^T Q_f B - K^T R &= (A - BK)^T Q_f B - K^T R \\
&= (A - B(R + B^T Q_f B)^{-1} B^T Q_f A)^T Q_f B - ((R + B^T Q_f B)^{-1} B^T Q_f A)^T R \\
&= A^T Q_f B - A^T Q_f B (R + B^T Q_f B)^{-1} B^T Q_f B - A^T Q_f B (R + B^T Q_f B)^{-1} R \\
&= A^T Q_f B \left[I - (R + B^T Q_f B)^{-1} (R + B^T Q_f B) \right] \\
&= A^T Q_f B \left[I - I \right] \\
&= 0
\end{aligned}$$

using $K = (R + B^T Q_f B)^{-1} B^T Q_f A$.

Then

$$J_k = \left\| x_{k+n_c-1} \right\|^2_{Q_f} + \left\| c_{k+n_c-1} \right\|^2_{B^T Q_f B + R} + \sum_{i=0}^{n_c-2} \left[\left\| x_{k+1+i} \right\|^2_Q + \left\| -K x_{k+i} + c_{k+i} \right\|^2_R \right]$$

Similarly for the next sample

$$\begin{aligned}
J_k = &\left\| x_{k+n_c-2} \right\|^2_{Q_f} + \left\| c_{k+n_c-1} \right\|^2_{B^T Q_f B + R} + \left\| c_{k+n_c-2} \right\|^2_{B^T Q_f B + R} \\
&+ \sum_{i=0}^{n_c-3} \left[\left\| x_{k+1+i} \right\|^2_Q + \left\| -K x_{k+i} + c_{k+i} \right\|^2_R \right]
\end{aligned}$$

consequently

$$J_k = \left\| x_k \right\|^2_{Q_f} + \sum_{i=0}^{n_c-1} \left\| c_{k+i} \right\|^2_{B^T Q_f B + R} \tag{3.26}$$

$$= \left\| x_k \right\|^2_{Q_f} + \sum_{i=0}^{n_c-1} \left\| c_{k+i} \right\|^2_S \tag{3.27}$$

where $\left\| x \right\|_A$ is defined as $\sqrt{x^T A x}$, $S = B^T Q_f B + R$ and $S > 0$. The term $\left\| x_k \right\|^2_{Q_f}$ in (3.27) does not depend on the decision variable and can be ignored in minimising this cost, giving

$$J_{c,k} = \sum_{i=0}^{n_c-1} \left\| c_{k+i} \right\|^2_S = \underset{\rightarrow}{c}_k^T S_c \underset{\rightarrow}{c}_k. \tag{3.28}$$

where $S_c = diag(S, \ldots, S)$.

If the constraints are expressed in terms of $\underset{\rightarrow}{c}_k$ then a quadratic programming (QP) can be formulated using dual mode predictions.

Algorithm 3.1. *Optimal MPC (OMPC) using closed loop paradigm*

1. *At each sampling instant, perform the optimisation:*

$$J_{c,k} = \sum_{i=0}^{n_c-1} \left\| c_{k+i} \right\|_S^2$$

Subject to

$$Mx_k + N\underset{\rightarrow}{c}_k \leq d.$$

for suitable M, N and d prediction matrices with finite number of rows.

2. *Implement the first component of $\underset{\rightarrow}{c}_k$, that is c_k in the control law of* (3.14).

3. *If unconstrained control law is satisfying the constraints (i.e. $x_k \in \mathcal{X}_0$),the optimising $\underset{\rightarrow}{c}_k$ is zero so the control law is $u_k = -Kx_k$.*

The weakness of OMPC is the trade off between MCAS volumes, that is the volume of MCAS $\mathcal{X}_c$ and the number of d.o.f. n_c; this is because the MAS can be quite small for well tuned feedback K and feasibility of the input parameterisation (3.14) require that one adopts the unconstrained law after n_c steps. Consequently, several authors have shown that well tuned optimal feedback gain K can result in a requirement for a large n_c in order to achieve reasonable MCAS volumes (e.g. [13]). A large n_c can imply a demanding optimisation, even with algorithms that exploit structure; this is one reason for the recent popularity of parametric methods [116].

Remark 3.2.2. It is well known that stability and recursive feasibility of OMPC can be shown using conventional arguments which deploy $J_{c,k}$ from (3.28) as a monotonically non-increasing function. The essence of the argument is that at sample instant $k+1$, one can re-use the remainder of the sequence $\underset{\rightarrow}{c}_k$ from the previous sample, and thus the new $J_{c,k}$ is upper bounded by a value that is smaller than at the previous sample, unless c_k is repeatedly zero. Once $\underset{\rightarrow}{c}_k = [0, \ldots, 0]$, note that all states inside the MAS (i.e. $x_k \in$ MAS), the unconstrained optimal law $u_k = -Kx_k$ will feasible. Hence, the

Laypunov stability of the origin follows from the fact that the MAS contains the origin in its interior.

3.3 Laguerre function parameterisation

This section gives a quick overview of how the associated Laguerre OMPC algorithm [26] was formulated.

3.3.1 Laguerre functions and predictions dynamics

Laguerre functions are defined as follows [18]

$$l_i(z) = \sqrt{(1-p^2)}\frac{(z^{-1}-p)^{i-1}}{(1-pz^{-1})^i} = l_{i,0} + l_{i,1}z^{-1} + \cdots; \quad 0 \leq p < 1. \tag{3.29}$$

where pole 'p' is the parameter which determines convergence rate. The corresponding sequences of terms $l_{i,k}$ for the first n functions can also be determined from an equivalent n-dimensional state-space model.

$$\begin{bmatrix} l_{1,k+1} \\ l_{2,k+1} \\ l_{3,k+1} \\ \vdots \end{bmatrix} = L_{k+1} = \underbrace{\begin{bmatrix} p & 0 & 0 & \dots \\ 1-p^2 & p & 0 & \dots \\ -p(1-p^2) & (1-p^2) & p & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{A_L} L_k,$$

$$L_0 = \sqrt{(1-p^2)}\begin{bmatrix} 1 & -p & p^2 & \dots \end{bmatrix}^T. \tag{3.30}$$

3.3.2 Laguerre optimal MPC (LOMPC)

Laguerre OMPC (LOMPC) is a dual mode MPC algorithm [15] where the input perturbation c_k are parameterised in terms of Laguerre functions. Input perturbations are

parameterised as follows:

$$\begin{bmatrix} c_k \\ c_{k+1} \\ \vdots \\ c_{k+n-1} \\ \vdots \end{bmatrix} = \begin{bmatrix} L_0^T \\ L_1^T \\ \vdots \\ L_{n-1}^T \\ \vdots \end{bmatrix} \begin{bmatrix} \eta_k \\ \eta_{k+1} \\ \vdots \\ \eta_{k+n-1} \\ \vdots \end{bmatrix} = \begin{bmatrix} L_0^T \\ L_0^T A_L^T \\ \vdots \\ L_0^T A_L^{n-1^T} \\ \vdots \end{bmatrix} \underset{\rightarrow}{\eta}_k = H_L \underset{\rightarrow}{\eta}_k \tag{3.31}$$

where η_k is a parameterised decision variable and it is noted that the perturbations converge to zero only asymptotically and thus this is a very different parameterisation to OMPC. The H_L matrix has a large number of rows (technically infinite) to capture the asymptotic behaviour whereas the number of columns represents the number of d.o.f.. Substituting input perturbations (3.31) using compatible dimensions into the cost function of (3.28) for an infinite n_c, the prediction cost can be represented in term of $\underset{\rightarrow}{\eta}_k$ as:

$$J_{L,k} = \underset{\rightarrow}{\eta}_k^T \left(\sum_{i=0}^{\infty} A_L^i L_0 S L_0^T A_L^{i\,T} \right) \underset{\rightarrow}{\eta}_k = \underset{\rightarrow}{\eta}_k^T S_L \underset{\rightarrow}{\eta}_k \tag{3.32}$$

with $c_{k+i} = L_i^T \underset{\rightarrow}{\eta}_k$ and $L_i = A_L L_{i-1}$. Constraints (or equivalently the new MCAS) are parameterised as

$$Mx_k + NH_L \underset{\rightarrow}{\eta}_k \leq d, \tag{3.33}$$

for appropriate M, N, H_L and d.

Algorithm 3.2. *Laguerre Optimal MPC (LOMPC)*

Off-line

1. *Determine the predicted cost, in terms of perturbations c_k as*

$$J_{c,k} = \sum_{i=0}^{\infty} c_{k+i}^T S c_{k+i}. \tag{3.34}$$

Substitute for infinite n_c from (3.30) *and* (3.31) *the LOMPC predictions of $c_{k+i} = L_i^T \underset{\rightarrow}{\eta}_k$ to give*

$$J_{L,k} = \sum_{i=0}^{\infty} \underset{\rightarrow}{\eta}_k^T L_i S L_i^T \underset{\rightarrow}{\eta}_k. \tag{3.35}$$

Finally, substitute $L_i = A_L L_{i-1}$ *and hence*

$$J_{L,k} = \underset{\rightarrow}{\eta}_k^T \left[\sum_{i=0}^{\infty} A_K^i L_0 S L_0^T (A_L^i)^T\right] \underset{\rightarrow}{\eta}_k = \underset{\rightarrow}{\eta}_k^T S_L \underset{\rightarrow}{\eta}_k. \tag{3.36}$$

2. *Define the constraint inequalities associated in the form*

$$Mx_k + NH_L \underset{\rightarrow}{\eta}_k \leq d. \tag{3.37}$$

<u>*On-line*</u>

1. *At each sampling instant, perform the optimisation:*

$$\begin{aligned} &\underset{\rightarrow}{\eta}_k^* = arg \min_{\underset{\rightarrow}{\eta}_k} \; J_{L,k} \\ &s.t.\; Mx_k + NH_L \underset{\rightarrow}{\eta}_k \leq d. \end{aligned} \tag{3.38}$$

2. *Define* $\underset{\rightarrow}{c}_k = [L_0^T, \ldots, L_{n_c-1}^T] \underset{\rightarrow}{\eta}_k^*$.

3. *Implement the first component of* $\underset{\rightarrow}{c}_k$*, that is* c_k *in the control law of* (3.14).

4. *If the unconstrained control law is satisfying the constraints (i.e.* $x_k \in \mathcal{X}_0$*),the optimising* $\underset{\rightarrow}{c}_k$ *is zero so the control law is* $u_k = -Kx_k$.

Within LOMPC, the user has a handle to trade off between MCAS volumes and the number of d.o.f., that is the parameter ' p'; although no generic guarantees can be given, examples have shown that in many cases slowing convergence by increasing 'p' above zero ($p = 0$ is equivalent to OMPC) can improve MCAS volumes, possibly at some expense to performance. Nevertheless, one key question was still left unanswered: is there a systematic way of choosing the best 'Laguerre function' or indeed is there an alternative to Laguerre which is better still? These questions are tackled in this thesis.

Remark 3.3.1. LOMPC has a guarantee of stability and recursive feasibility, in the nominal case [138, 187]. This follows the conventional arguments Section 3.1.4, [6, 10, 11] whereby one can show that the cost function $J_{L,k}$ is monotonically non-increasing so long as the optimisation at sample k reuse the trajectory computed at the previous sample. Finally, note that for all states inside the MAS, the unconstrained optimal control law

$u_k = -Kx_k$ will feasible, i.e. $\underset{\rightarrow}{c}_k = [L_0^T, \ldots, L_{n_c-1}^T]\underset{\rightarrow}{\eta}_k = 0$. Hence, the Laypunov stability of the origin follows from the fact that the MAS contains the origin in its interior. Recursive feasibility uses the similar argument.

3.4 Triple mode MPC

Predictive control has been developed widely both in the process industry and control research community and has reached a high level of maturity in its linear variant. A major remaining obstacle is to balance, for given n_c: (i) the desirable volume of maximal region of attraction $\mathcal{X}_c$ with (ii) complexity and hence the available computational power as well as (iii) achievable performance.

- If n_c is large enough, one can show that $\mathcal{X}_c$ is the largest feasible invariant set possible and moreover the control law is the global optimum [11].
- In general, for computational (and sometimes robustness) reasons, n_c is chosen small.
- If n_c is small, then the volume of $\mathcal{X}_c$ maybe dominated by the implied state feedback K, hence a highly tuned (in terms of performance) K could give rise to small $\mathcal{X}_c$ and a lesser tuned K could give much larger $\mathcal{X}_c$.
- Conversely, if K is poorly tuned, then the cost function is dominated by poorly performing predictions and the closed loop control may also be severely suboptimal.

The designer has to get a balance between the volume of the maximal region of attraction (affected by K and n_c), the computational load (implied by n_c) and the implied performance (affected by K and n_c). One suggestion that has been little considered in the literature is the concept of triple mode control [113]. In this strategy one recognises that large regions of attraction in conjunction with good performance often implies nonlinear or linear time varying (LTV) prediction dynamics [176].

So, instead of the dual mode predictions structure of (3.14), some authors have proposed terminal controls such as

$$\begin{aligned}
&\text{Mode 1} \quad u_{k+i} = -Kx_{k+i} + q_{k+i} + c_{k+i}, && i = 0, \ldots, n_c - 1 \\
&\text{Mode 2} \quad u_{k+i} = -Kx_{k+i} + q_{k+i}, && i = n_c, \ldots, n_c + n_q - 1 \\
&\text{Mode 3} \quad u_{k+i} = -Kx_{k+i}, && i \geq n_c + n_q
\end{aligned} \tag{3.39}$$

where the notable change is the introduction of terms q_i, $i = 0, \ldots, n_q - 1$ and hence the addition of a 3rd mode into the prediction control law. Here, c_i are the d.o.f. to be optimised online, whereas, ideally, the q_i could be inferred online based on offline or previous optimisations (it is mentioned in [114] that occasionally this needs to be reseeded). For example, in [113, 188], the second model control moves are defined as

$$q = \left[q_0^T, \ldots, q_{n_q-1}^T\right] = Hx_{n_c} \tag{3.40}$$

that is, the q_i values depend only on the predicted state at the commencement of mode 2. Then, with trivial algebra (simulating the model (3.1) with (3.39) and (3.40)), one can show that the Mode 2 predictions take the form of LTV feedback,

$$u_k = -K_{k-n_c} x_k, \qquad k = n_c, \ldots, n_c + n_q - 1, \tag{3.41}$$

where K_i depend on K, H, A and B.

The cost function J_k for the triple mode predicted feedback structure can be written as

$$J_k = \underrightarrow{c}_k^T S_T \underrightarrow{c}_k + \underrightarrow{c}_k^T V_T x_k + p_T, \tag{3.42}$$

for suitable S_T, V_T and p_T [113] with constraint of the form

$$M_T x_k + N_T \underrightarrow{c}_k \leq d_T \tag{3.43}$$

The matrix H is chosen such that it implies a maximal feasible invariant set. The next section will discuss earlier ellipsoidal based algorithms developed for selecting the best H to use in triple mode MPC.

Triple mode MPC using Ellipsoidal Invariant sets

Early triple mode MPC algorithms were motivated by the robust case and thus began with the work of [79] and ellipsoidal invariant sets, e.g.

$$\mathcal{X}_E = \{x \mid x^T P_e x \leq 1\}; \quad P_e > 0 \tag{3.44}$$

where $\Phi^T P_e \Phi \preceq P_e$ and constraints (3.3) are always satisfied with the control law $u_k = -Kx_k$. However, ellipsoidal invariant sets are conservative in volume and thus give artificially tight limits on feasibility; points outside the set may still be feasible. Within triple mode algorithms, the ellipsoidal sets are used as a systematic but interim step to finding a suitable H and are not deployed in the final algorithm.

Dual mode control is so effective because one is able to make implicit assumptions on the terminal mode and hence only compute the initial mode explicitly using polytopic constraints. Similarly, to form an efficient triple mode algorithm, it is necessary to make implicit assumptions for the terminal mode and mode 2 while selecting the initial mode explicitly using polytopic constraints.

The offline problem of ERPC [21] and GERPC [24] can be used to specify the second mode control moves for linear time invariant triple mode MPC, that is finding the matrix H in (3.40).

The invariant set $\mathcal{E}_z$ in $z_k = [x_k^T \quad \underrightarrow{q}_k^T]^T$ – space is defined as

$$\begin{bmatrix} x_k^T & \underrightarrow{q}_k^T \end{bmatrix} \underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}}_{Q_z^{-1}} \begin{bmatrix} x_k^T \\ \underrightarrow{q}_k^T \end{bmatrix} \leq 1. \tag{3.45}$$

Now, the projection onto x-space is given by $\mathcal{E}_x = \{x_k : \exists \ \underrightarrow{q}_k \ s.t. \ x_k^T P_{11} x_k \leq 1 - \underrightarrow{q}_k^T P_{22} \underrightarrow{q}_k - 2x_k^T P_{12}^T \underrightarrow{q}_k\}$. It is maximised when $\underrightarrow{q}_k = -P_{22}^{-1} P_{21} x_k$ and for nominal case, $-P_{22}^{-1} P_{21}$ is a possible choice for H, as first suggested in [113].

Before discussing the robust predictive control, the following definition summarise the concept of polyhedral set:

Definition 3.5 (Polyhedral Set)**.** *A set $\mathcal{P} \subset \mathbb{R}^n$ is a polyhedron if there is a system of finite many inequalities $Mx_k \leq d$ such that $\mathcal{P} = \{x_k \in \mathbb{R}^n \mid Mx_k \leq d\}$.*

3.5 Robust predictive control

Largely, one can divide the literature on robust linear MPC into two groups, based on which uncertainty class they study: Either LTI systems with unknown, bounded disturbances (see e.g. [82, 189]), or, linear systems with polytopic uncertainty, as which will be considered in this thesis (see e.g. [21, 79]). Another important feature distinguishing different approaches to robust MPC, is which class of online optimisations they utilise.

In robust MPC, the closed-loop paradigm improves the control performance and computational burden [83,105,115]. Reduced-complexity invariant sets were introduced in [115] for the case of quasi-infinite horizon closed loop MPC. The reduced-complexity invariant sets may result in a decrease in the number of online optimisation variables [115]. This invariant set structure is used in the design of [83, 167, 190] robust MPC and this thesis pursues this type of approach to including uncertainty information in the model.

Problem formulation for robust MPC

This thesis considers a time linear parameter varying (LPV) state space model of the form:

$$x_{k+1} = A_k x_k + B_k u_k, \quad k \in \mathbb{N} \tag{3.46}$$

with polytopic uncertainty description (see [169, 191])

$$[A_k, B_k] \in Co\{[A_1, B_1], ..., [A_m, B_m]\}, \tag{3.47}$$

or equivalent

$$[A_k, B_k] \in \left\{ \sum_{j=1}^{m} \lambda_j [A_j, B_j] \mid \lambda_1(k) \geq 0, \ldots, \lambda_m(k), \sum_{j=1}^{m} \lambda_j(k) = 1 \right\}, \ k \in \mathbb{N}. \tag{3.48}$$

Assume that the states and inputs at all time instants should fulfill the following constraints (the mixed state and input constraints are considered in line with the proposals in [25] from (3.2)):

$$L_x x + L_u u \leq l. \tag{3.49}$$

Next section reviews robust dual mode algorithm followed by a triple mode variant for

the uncertain case.

3.6 Robust dual mode MPC algorithms

The robust mode approaches model will use the dual mode prediction structure (3.14), where K should be stabilising for (3.46), and preferably optimising in some sense (for instance unconstrained optimal for some nominal model). Algorithms are based on both ellipsoidal and polyhedral invariant sets.

3.6.1 Robust dual mode MPC based on invariant polyhedral set

Typical robust MPC algorithms which deal with this case suffer a number of limitations. It is generally assumed that one cannot form the entire class of predictions due to the combinatorial explosion in the required number of terms [83, 192]. As a consequence the vast majority of work has made use of low complexity invariant sets, in particular ellipsoidal sets [79] or very simple polyhedral sets [193]. But the use of low complexity invariant sets implies a corresponding restriction to the region of attraction. Some authors tackled this limitation by augmenting the state dimension [21, 194] and in effect allowing a time varying control strategy. However, the inherent restriction on ellipsoidal sets remains in those work.

Polyhedral invariant set for LPV systems

One can extend nominal MPC to the robust case if one can determine a suitable feasible invariant polyhedral set; such work was given [115] for the standard dual mode algorithm. The key idea used is not dissimilar to the one step sets popularised in [195], that is to use backwards predictions rather than forwards prediction. This simple change eliminates the combinatorial explosion in the possible number of predictions terms and hence creates a tractable problem. A feasible invariant polyhedral set can be computed using the algorithm proposed in [115].

Robust MPC using polyhedral invariant set

It was shown that [83], subject to a quadratic stabilising criterion, the robust maximal feasible invariant set $\mathcal{X}_r$ takes the same form as $\mathcal{X}_c$, but different matrices. i.e.

$$M_r x_k + N_r \underset{\rightarrow}{c}_k \leq d_r. \tag{3.50}$$

An algorithm was introduced in [115], that allows the construction of polyhedral invariant sets for LPV systems with polytopic uncertainty description. The algorithm iteratively adds constraints to the imposed constraint sets until robust invariance is obtained. Due to the fact that only non-redundant constraints are added and that redundant constraints are regularly removed, a large efficiency increase can be obtained compared to other algorithms. This algorithm is used to calculate the polyhedral invariant set for LPV systems and it avoids the combinational explosion in the number of terms.

After computing the polyhedral set, the online optimisation consists of minimising the upper bound on the infinite horizon cost. A predicted cost can be constructed as a quadratic function i.e. $\underset{\rightarrow}{c}_k^T S_c \underset{\rightarrow}{c}_k$ [114, 188]. It can be shown [22] that $\underset{\rightarrow}{c}_k^T S_c \underset{\rightarrow}{c}_k$ and the infinite cost differ by a bias term, thus minimising the two indices is equivalent [24].

At each sample solve the following optimisation problem

$$\begin{aligned} &\min_{\underset{\rightarrow}{c}_k} \ \underset{\rightarrow}{c}_k^T S_c \underset{\rightarrow}{c}_k \\ s.t. \ \ &M_r x_k + N_r \underset{\rightarrow}{c}_k \leq d_r. \end{aligned}$$

Use the first block element of $\underset{\rightarrow}{c}_k$ in the control law $u_k = -Kx_k + c_k$.

3.6.2 Robust dual mode MPC based on invariant ellipsoids

The idea of augmenting the system dynamics with the 'n_c' future d.o.f. was proposed in [21]. By doing this, feasibility in robust dual mode MPC could be handled offline by optimising the size of an ellipsoid subject to constraints and invariance for the augmented dynamics. A convex SDP problem, denoted as generalised ERPC (GERPC), is formulated in [25] with ERPC proposed in [21] as a special case.

Generalised Efficient Robust Predictive Control (GERPC)

Consider the autonomous state space model created by augmenting the state vector with the future 'd.o.f.' [21]. Letting $\underset{\rightarrow}{c}_k$ denote the vector of future perturbations away from optimal control, over the control horizon, n_c, the dynamics of (3.46) can be described as

$$z_{k+1} = \psi_k z_k,\ z_k = \begin{bmatrix} x_k \\ \underset{\rightarrow}{c}_k \end{bmatrix}$$
$$\psi_k \in Co\{\psi_j,\ j = 1\ldots, m\},\ \psi_j = \begin{bmatrix} \Phi_i & B_i D \\ 0 & G_c \end{bmatrix}, \tag{3.51}$$

where $z \in \mathbb{R}^{n_x + n_u n_c}$, $\underset{\rightarrow}{c}_k^T = [c_k^T, c_{k+1}^T, \ldots, c_{k+n_c-1}^T]$, D and G_c are variables that are used to optimise size and shape of the associated feasible invariant ellipsoid. ERPC used $G_c = I_L$, $D = E$ and

$$\Phi_i = A_i - B_i K,\ E = \begin{bmatrix} I_{n_u}, 0, \ldots, 0 \end{bmatrix},\ I_L = \begin{bmatrix} 0_{n_u} & I_{n_u} & 0_{n_u} & \ldots & 0_{n_u} \\ 0_{n_u} & 0_{n_u} & I_{n_u} & \ldots & 0_{n_u} \\ & \vdots & & & \\ 0_{n_u} & 0_{n_u} & \ldots & 0_{n_u} & I_{n_u} \\ 0_{n_u} & 0_{n_u} & \ldots & 0_{n_u} & 0_{n_u} \end{bmatrix}. \tag{3.52}$$

Here z_k is the augmented state vector, ψ is the augmented transition matrix, I_{n_u} is the identity matrix, and 0_{n_u} is a matrix of zeros, both of dimension $n_u \times n_u$. The I_L matrix simply accommodates the time recession of $\underset{\rightarrow}{c}_k$. Note that the structure of I_L ensures

$$c_{k+n_c+i} = 0\ \ \forall i \geq 0. \tag{3.53}$$

The associated constraints (3.49) $\forall k$ are represented as:

$$\begin{bmatrix} L_x - L_u K & L_u D \end{bmatrix} z_k \leq l. \tag{3.54}$$

In GERPC, $\underset{\rightarrow}{f}_k$ defines future control perturbations through the dynamics $u_k = -Kx_k + D\underset{\rightarrow}{f}_k$, $\underset{\rightarrow}{c}_k = D\underset{\rightarrow}{f}_k$ $\underset{\rightarrow}{f}_{k+1} = G_c \underset{\rightarrow}{f}_k$.

D and G_c are chosen to ensure robust invariance of the ellipsoid $\mathcal{E}_z = \left(z : z^T Q_z^{-1} z \leq 1\right)$

in conjunction with maximising the size of projection of $\mathcal{E}_z$ to x-space, $\mathcal{E}_x$:

$$\begin{aligned}
&\min_{Q_z,D,G_c} \; ln\; det(TQ_zT^T)^{-1} \\
&\quad s.t. \;\; \begin{bmatrix} \Phi_j & B_jD \\ 0 & G_c \end{bmatrix}^T Q_z^{-1} \begin{bmatrix} \Phi_j & B_jD \\ 0 & G_c \end{bmatrix} - Q_z^{-1} \\
&\qquad < -\frac{1}{\gamma} \begin{bmatrix} I & 0 \\ -K & D \end{bmatrix}^T \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} I & 0 \\ -K & D \end{bmatrix}, j = 1, \ldots, m. \\
&\qquad \begin{bmatrix} W & [L_x - L_uK \quad L_uD] \\ [L_x - L_uK \quad L_uD]^T & Q_z^{-1} \end{bmatrix} > 0, W_{ii} \leq l_i^2.
\end{aligned} \tag{3.55}$$

This problem can be solved by SDP packages [196], where γ provides a tuning parameter for the size of region of attraction verses online cost trade off for GERPC [25].

Robust MPC using invariant ellipsoids

Given design parameters n_c, Q, R and γ, calculate D, G_c and Q_z from (3.55). The predicted cost can be constructed as $\underset{\rightarrow}{c}_k^T \Gamma \underset{\rightarrow}{c}_k$ where Γ is the positive definite solution of the discrete Lyapunov equation $G_c^T \Gamma G_c - \Gamma = -D^T S_c D$. This turns the online problem into minimising a quadratic function subject to ellipsoidal constraints.

At each sample solve the following optimisation problem

$$\begin{aligned}
&\min_{\underset{\rightarrow}{c}_k} \; \underset{\rightarrow}{c}_k^T \Gamma \underset{\rightarrow}{c}_k \\
&s.t. \;\; z^T Q_z z \leq 1.
\end{aligned}$$

Use the first block element of $\underset{\rightarrow}{c}_k$ in the control law $u_k = -Kx_k + c_k$.

3.7 Robust triple mode MPC

This section shows how the solutions of a suitable robust G(ERPC) offline problem can be used to specify a robust triple mode MPC algorithm deploying polyhedral sets.

$$\begin{aligned}
&\text{Mode 1} \quad u_{k+i} = -Kx_{k+i} + q_{k+i} + c_{k+i}, && i = 0, \ldots, n_c - 1, \\
&\text{Mode 2} \quad u_{k+i} = -Kx_{k+i} + q_{k+i}, && i = n_c, \ldots, n_c + m_c - 1, \\
&\text{Mode 3} \quad u_{k+i} = -Kx_{k+i} && i \geq n_c + m_c
\end{aligned} \tag{3.56}$$

where c_k will be the d.o.f. chosen directly and q_k will define from the (G)ERPC offline solution as $\underset{\rightarrow}{q}_k = D\underset{\rightarrow}{f}_k$, $\underset{\rightarrow}{f}_{k+1} = G_c\underset{\rightarrow}{f}_k$ [114, 188]. The prediction dynamics are defined as [114, 188]

$$X_{k+1} = \Psi_k X_k, \quad X_k = \begin{bmatrix} x_k \\ \underset{\rightarrow}{f}_k \\ \underset{\rightarrow}{c}_k \end{bmatrix}, \tag{3.57}$$
$$\Psi_k \in Co\{\Psi_j,\ j = 1\ldots, m\},$$

with I_L as shift matrix. These dynamics should fulfill the constraints given by

$$\begin{bmatrix} L_x - L_u K & L_u D & L_u E \end{bmatrix} X_k \leq l, \quad \forall k \tag{3.58}$$

uncertain description of augmented dynamics is given by [114, 188]

$$\Psi_j = \begin{bmatrix} A_j - B_j K & B_j D & B_j E \\ 0 & G_c & 0 \\ 0 & 0 & I_L \end{bmatrix}. \tag{3.59}$$

3.7.1 Cost for triple mode predictions

A predicted cost $J(x_k, \underset{\rightarrow}{f}_k, \underset{\rightarrow}{c}_k)$ that bounds the infinite horizon cost of (3.57) initialised at $(x_0, \underset{\rightarrow}{f}_0, \underset{\rightarrow}{c}_k)$ (that is, the infinite horizon cost of (3.46) using the prediction setup (3.56)) should be found:

$$J(x_0, \underset{\rightarrow}{f}_0, \underset{\rightarrow}{c}_k) \geq \max_{[A_k,B_k]\in Co\{[A_{1,k},B_{1,k}],\ldots,[A_{m,k},B_{m,k}]\}} \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k. \tag{3.60}$$

Such a cost can be constructed as [114, 188]

$$J(x_k, \underset{\rightarrow}{f}_k, \underset{\rightarrow}{c}_k) = [x_k\ \underset{\rightarrow}{f}_k\ \underset{\rightarrow}{c}_k]^T P [x_k\ \underset{\rightarrow}{f}_k\ \underset{\rightarrow}{c}_k] \tag{3.61}$$

where $P > 0$ satisfies

$$P - \Psi_j^T P \Psi_j \geq [I\ 0\ 0]^T Q [I\ 0\ 0]$$
$$+[-K\ D\ E]^T R [-K\ D\ E], \quad j = 1, ..., m. \tag{3.62}$$

The matrix P can be efficiently calculated by the SDP

$$\min_{P} \; tr(P) \quad s.t. \quad (3.62). \tag{3.63}$$

3.7.2 The robust triple mode algorithm

Given designed parameters n_c, Q, R and γ, calculate first D and G_c from (3.55), then M, N_1, N_2 and d using algorithm in [115], and finally P from (3.63).

At each sample solve the following optimisation problem

$$\begin{aligned} &\min_{\underrightarrow{c}_k} \; J(x_k, \underrightarrow{f}_k, \underrightarrow{c}_k) \\ s.t. \;\; &Mx_k + N_1 \underrightarrow{f}_k + N_2 \underrightarrow{c}_k \leq d. \end{aligned}$$

for appropriate M, N_1, N_2, d. Implement $u_k = -Kx_k + D\underrightarrow{f}_k + E\underrightarrow{c}_k$ to the plant. Calculate $\underrightarrow{f}_{k+1} = G_c \underrightarrow{f}_k$.

3.8 Summary

This section summaries the key point and discusses the limitation in already proposed algorithms.

Optimal MPC

The weakness of OMPC is the trade off between MCAS volumes, that is the volume of MCAS $\mathcal{X}_c$ and the number of d.o.f. n_c; this is because the MAS can be quite small for well tuned feedback K and feasibility of the input parameterisation (3.14) requires that one adopts the unconstrained law after n_c steps. Consequently, several authors have shown that well tuned optimal feedback gain K can result in a requirement for a large n_c in order to achieve reasonable MCAS volumes (e.g. [13]).

Laguerre OMPC

In LOMPC, the user has a handle to trade off between MCAS volumes and the number of d.o.f., that is the parameter ' p'; although no generic guarantees can be given, examples

have shown that in many cases slowing convergence by increasing 'p' above zero ($p = 0$ is equivalent to OMPC) can improve MCAS volumes, possibly but not necessarily at some expense to performance. Nevertheless, one key question was still left unanswered: is there a systematic way of choosing the best 'Laguerre function' or indeed is there an alternative to Laguerre which is better still?

Triple mode MPC

The concept of triple mode control was proposed to overcome the conflict between performance and the region of attraction. The main weakness in triple mode MPC is linked to the efficiency of the middle mode, can this be computed implicitly or explicitly and also is the offline optimisation for identifying a suitable G_c overly complex? Should polyhedral sets or ellipsoidal sets be deployed to define the middle mode?

3.9 Conclusion

This chapter provided a common theoretical background necessary for arguments in this thesis. Constrained stable MPC was formulated by a dual mode scheme in the predictions, whereby the first mode can be used for constraint handling with the constraint that the state at the end of the control horizon enters a terminal invariant set, commonly chosen to be the MAS to maximise the solution space, which has the property that constraints are not violated in this set for a given terminal control law. An optimal MPC algorithm was formulated using dual mode predictions, Laguerre function parameterisations within an optimal MPC, and triple mode approaches using ellipsoidal and polyhedral sets. Thereafter robust MPC was formulated for both dual mode and triple mode approaches using a linear parameter varying system. Finally, it was highlighted that there is a well understood trade off between the region of attraction, performance and computational burden within already proposed algorithms.

Part II

Generalised function parameterisations within model predictive control

Chapter 4

Kautz functions to enlarge the region of attraction within predictive control

This chapter presents an **original contribution** to the thesis. It develops the recently published Laguerre optimal model predictive control by proposing a more flexible parameterisation of the degrees of freedom in order to further increase the region of attraction of model predictive control (MPC). Specifically, a simple but efficient algorithm that uses Kautz functions to parameterise the degrees of freedom in Optimal MPC is presented. It is shown that this modification gives a mechanism to achieve low computational burden with enlarged region of attraction and improved performance. The improvement, with respect to the existing algorithm that uses a Laguerre parameterisation [18, 26], is demonstrated by examples. It is also shown that the proposed algorithms have standard convergence and feasibility guarantees.

The chapter is organised as follows: Section 4.1 presents the introduction and motivation of the chapter; Section 4.2 presents the necessary background on Optimal MPC formulation and Laguerre optimal MPC; Section 4.3 presents the basic properties of Kautz functions and compares them with Laguerre functions; Section 4.4 develops the novel Kautz OMPC (KOMPC) algorithm using Kautz functions to parameterise the input predictions; Section 4.5 presents numerical examples; and finally, Section 4.6 gives the conclusion of the chapter.

4.1 Introduction

One key conflict in linear predictive control is that between a region of attraction and performance. If a dual-mode MPC controller, that is tuned to give high performance, it will often have relatively small regions of attraction [11,13] unless one uses a prohibitively large number of decision variables (or degrees of freedom, d.o.f.). There is a pragmatic limit to increase the d.o.f. for the global region of attraction as this compromises the computational burden. A strategy with the same number of d.o.f. giving a larger region of attraction might be achieved through detuning of terminal mode but hence has relatively worse performance [113].

Laguerre functions in Section 3.3 have been proposed as a means of parameterising the input predictions in [26] as a simple way of improving the performance and region of attraction. The main idea is to form the predictions as a combination of Laguerre functions. Specifically it was shown that in many cases changing the parameterisation allowed substantial improvements in region of attraction with little or no detriment to performance. Nevertheless, one key question was still left unanswered: is there an alternative to Laguerre function or is this the only choice?

This chapter assumes that the terminal mode is well tuned and therefore this parameter is not available for influencing the size of the region of attraction. Thus the only way to enlarge the region attraction is with d.o.f. within or parameterisation of the predictions. Specifically the intent is to answer the question about Laguerre OMPC, that is, how else can the designers increase the region of attraction or indeed is there an alternative to Laguerre function which is better still? Hence, in line with the proposals of [26], here Kautz functions are tested as these are more flexible than Laguerre functions. Kautz functions are also a special case of generalised prediction framework in [25] with a lower triangular structure as already discussed for Laguerre functions in Section 2.6.5. This chapter will demonstrate that Kautz functions are an effective alternative to the standard basis set for parameterising the d.o.f. within MPC and indeed may be more effective that Laguerre functions as they offer more variety in the key characteristics.

4.2 Background

This section will introduce the basic algorithms [11, 13], underneath the proposal in the chapter and background information, but omitting well known algebra that does not add to the concepts. The main aspects of formulating a linear MPC problem as a Laguerre

function parameterisation will be repeated here for convenience (see Section 3.2 for further details).

4.2.1 Optimal MPC

Consider a generic MPC problem formulation [11, 13]

$$\begin{aligned} \min_{u_k} x_{n_c}^T Q_f x_{n_c} + \sum_{k=0}^{n_c-1} x_k^T Q x_k + u_k^T R u_k \\ s.t.\ \ x_{k+1} = Ax_k + Bu_k, \\ x_k \in \mathbb{X},\ \ u_k \in \mathbb{U}, \\ x_{n_c} \in \mathcal{X}_0, \end{aligned} \tag{4.1}$$

with $Q = Q^T \geq 0$ and $R = R^T > 0$ are state and input cost weighting matrices and $\mathcal{X}_0 = \{x_k \in \mathbb{R}^{n_x} | \Phi^i x_k \in \mathbb{X},\ -K\Phi^i x_k \in \mathbb{U}, \forall i \geq 0\}$ [12] is a terminal region. Assuming that state-feedback $u_k = -Kx_k$ is the optimal unconstrained feedback as $n_c \to \infty$, then with appropriate choice of Q_f $\mathcal{X}_0$ the MAS to match this K, optimisation (4.1) defines the Optimal MPC (OMPC) algorithm [5, 11, 13] which is able to match the constrained optimal control, as long as n_c is large enough. However, for transparency, it is common to emphasise that the associated input predictions take the following parameterisation where vector $\underset{\rightarrow}{c}_k = [c_k^T, \ldots, c_{k+n_c-1}^T]$ compromise the d.o.f.; in essence $\underset{\rightarrow}{c}_k$ are the perturbations about the unconstrained optimal input trajectory required to meet constraints

$$\begin{aligned} u_{k+i} &= -Kx_{k+i} + c_{k+i}, & i &= 0, \ldots, n_c - 1, \\ u_{k+i} &= -Kx_{k+i}, & i &\geq n_c. \end{aligned} \tag{4.2}$$

An equivalent compact formulation for OMPC with suitable S, M, N and d prediction matrices and vectors [5] is

$$\begin{aligned} \min_{\underset{\rightarrow}{c}_k} \ \ & \underset{\rightarrow}{c}_k^T S \underset{\rightarrow}{c}_k \\ s.t.\ \ & Mx_k + N\underset{\rightarrow}{c}_k \leq d. \end{aligned} \tag{4.3}$$

The weakness of OMPC is the trade off between a region of attraction, that is the volume of the MCAS $\mathcal{X}_c = \{x_k \in \mathbb{R}^{n_x} | \exists \underset{\rightarrow}{c}_k \in \mathbb{R}^{n_c n_u}, Mx_k + N\underset{\rightarrow}{c}_k \leq d\}$ and the number of d.o.f. n_c; this is because the MAS can be quite small for well tuned feedbacks K and region of attraction of the input paramaterisation (4.2) requires that one adopts the unconstrained

law after n_c steps. Consequently, several authors have shown that well tuned optimal feedback gain K can result in a requirement for a large n_c in order to achieve reasonable volumes of region of attraction (e.g. [13]). A large n_c can imply a demanding optimisation, even with algorithms that exploit structure; this is one reason for the recent popularity of parametric methods [118].

4.2.2 Laguerre optimal MPC

Laguerre functions in Section 3.3 were used to provide a possible solution to the MCAS volume/complexity trade off in OMPC [26]. Conceptually the key point is to parameterise the future values of input perturbations $\underset{\rightarrow}{c}_k$ in (4.2) in terms of Laguerre functions. Specifically it was shown that in many cases changing parameterisation allowed substantial improvements in the region of attraction with little or no detriment to performance [26]. The input perturbations are parameterised as follows

$$\begin{aligned} \underset{\rightarrow}{c}_k &= [L_0^T, \ldots, L_0^T {A_L^{n_c-1}}^T, \ldots]^T \underset{\rightarrow}{\eta}_k, \\ \underset{\rightarrow}{\eta}_{k+1} &= H_L \underset{\rightarrow}{\eta}_k, \quad H_L = [L_0^T, \ldots, L_0^T {A_L^{n_c-1}}^T, \ldots]^T \end{aligned} \tag{4.4}$$

where A_L from (3.31) is a Laguerre dynamic matrix, hence an equivalent compact formulation similar to (4.3) with suitable prediction matrices and vectors is

$$\begin{aligned} \min_{\underset{\rightarrow}{\eta}_k} \quad & \underset{\rightarrow}{\eta}_k^T [\sum_{i=0}^{\infty} A_L^i L_0 S L_0^T {A_L^i}^T] \underset{\rightarrow}{\eta}_k \\ s.t. \quad & M_L x_k + N_L H_L \underset{\rightarrow}{\eta}_k \leq d_L. \\ & H_L = [L_0^T, \ldots, L_0^T {A_L^{n_c-1}}^T, \ldots]^T. \end{aligned} \tag{4.5}$$

This section summarised algorithms already in the literature which introduced the concept of reparameterisating the input trajectory d.o.f., but did not explore this potential in any depth. This chapter will further explore more flexible parameterisation techniques based on Kautz functions.

4.3 Kautz Functions

Laguerre functions [2, 18, 26] have been popular in filtering, system identification and control design because few parameters are enough to describe the behaviour of the sys-

tem. Its property depends upon the selection of a scaling factor (i.e. 'p' in (4.6)) and its complexity and accuracy of the description increases as the number of Laguerre networks increases. The application of Laguerre networks [2, 18] is limited to a single pole selection (i.e. between 0-1). This limitation may be overcome by introducing Kautz networks which allow the selection of two poles (real or complex) which consequently may approximate system behaviour better than using a single real pole as in Laguerre networks.

The Kautz function model was used for system identification of lightly damped systems to design robust predictive control [197,198]. A predictive functional control strategy based on the incremental Kautz model has been presented in [199], which does not require a correct model of the system before undertaking control design. In [200], five stochastic search algorithms were designed to optimise the adaptive parameter in predictive functional controller based on Kautz model. The continuous-time Laguerre functions and Kautz functions were discussed in [2] and were utilized in the design of continuous time model predictive control. In this chapter Kautz functions are used to parameterise the input trajectory to simplify the trade off between performance, region of attraction and computational burden.

4.3.1 Kautz Network

Kautz networks were first proposed by Kautz [201]. The discrete time Kautz network was generated from the discretisation of continuous time Kautz network (a more detailed discussion on continuous time Kautz functions can be found in [18]).

The Laguerre network has a first order dynamics where as the Kautz network has second order dynamics. Laguerre functions are defined as follows [18]

$$l_i(z) = \sqrt{(1-p^2)}\frac{(z^{-1}-p)^{i-1}}{(1-pz^{-1})^i}; \qquad 0 \leq p < 1. \tag{4.6}$$

where pole 'p' is the parameter which determines the convergence rate.

The Kautz network is defined as follows

$$k_i(z) = \sqrt{(1-a^2)(1-b^2)}\frac{(z^{-1}-a)^{i-1}(z^{-1}-b)^{i-1}}{(1-az^{-1})^i(1-bz^{-1})^i}; \tag{4.7}$$
$$0 \leq a < 1; \;\; 0 \leq b < 1$$

where 'a' and 'b' are poles of the discrete-time Kautz network. The free parameters, 'a'

and 'b' is selected by the user; these are also called the scaling factors. However, the inverse z-transforms of the Kautz networks do not lead to a compact expression of the Kautz functions in the time-domain so state-space representation is preferred and derived briefly.

4.3.2 Orthonormality of Kautz functions

The z-transforms of the discrete-time Kautz functions are written as

$$\begin{aligned} k_1(z) &= \frac{\sqrt{(1-a^2)(1-b^2)}}{(1-az^{-1})(1-bz^{-1})} \\ k_2(z) &= \frac{\sqrt{(1-a^2)(1-b^2)}}{(1-az^{-1})(1-bz^{-1})}\frac{(z^{-1}-a)(z^{-1}-b)}{(1-az^{-1})(1-bz^{-1})} \\ &\vdots \\ k_n(z) &= \sqrt{(1-a^2)(1-b^2)}\frac{(z^{-1}-a)^{n-1}(z^{-1}-b)^{n-1}}{(1-az^{-1})^n(1-bz^{-1})^n} \end{aligned} \tag{4.8}$$

where $0 \leq (a, b) < 1$ for stability of the functions. The Kautz functions are well known for their orthonomality. In the frequency domain, this orthonormality is expressed in terms of the orthonormal equations for $k_m(m = 1, 2, \dots)$ as

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} k_m(e^{jw})k_n(e^{jw})^* dw = 1; \;\; m = n \tag{4.9}$$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} k_m(e^{jw})k_n(e^{jw})^* dw = 0; \;\; m \neq n \tag{4.10}$$

where $(.)^*$ denotes the complex conjugate of $(.)$.

4.3.3 State space representation of Kautz functions

In the design of predictive control, Kautz functions are used in time domain. The discrete-time Kautz functions are obtained through the inverse z-transform of the Kautz discrete network. However, taking the inverse z-transform does not lead to a compact expression of the Kautz functions in the time domain. A more straightforward way to find these

discrete-time functions is based on a state space realisation of the polynomials.

$$k_i(z) = k_{i-1}(z)\frac{(z^{-1}-a)(z^{-1}-b)}{(1-az^{-1})(1-bz^{-1})}; \qquad 0 \le a < 1;\ \ 0 \le b < 1 \tag{4.11}$$

with $k_1(z) = \frac{\sqrt{(1-a^2)(1-b^2)}}{(1-az^{-1})(1-bz^{-1})}$. The discrete-time Kautz functions are expressed in a vector form as

$$k_n = [k_{n,1}, k_{n,2}, \dots]^T. \tag{4.12}$$

Taking advantage of the network realization in equation (4.11), the set of discrete-time Kautz functions satisfies the following difference equation

$$\mathcal{K}_{k+1} = A_K \mathcal{K}_k \tag{4.13}$$

where matrix size of A_K is $N \times N$ and is a function of parameters $\alpha_1 = ab$, $\beta_1 = (1-ab)$ and $\varsigma = \sqrt{(1-a^2)(1-b^2)}$, and the initial condition is given by $\mathcal{K}_0$ (see Section A.1 for further details), which yields to

$$\begin{bmatrix} k_{1,k+1} \\ k_{2,k+1} \\ k_{3,k+1} \\ k_{4,k+1} \\ k_{5,k+1} \\ \vdots \end{bmatrix} = \mathcal{K}_{k+1} = \underbrace{\begin{bmatrix} b & 0 & 0 & 0 & 0 & \dots \\ \beta_1 & a & 0 & 0 & 0 & \dots \\ -b\beta_1 & \beta_1 & b & 0 & 0 & \dots \\ \alpha_1\beta_1 & -b\beta_1 & \beta_1 & a & 0 & \dots \\ -b\alpha_1\beta_1 & \alpha_1\beta_1 & -b\beta_1 & \beta_1 & b & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{A_K} \mathcal{K}_k; \quad \mathcal{K}_0 = \varsigma \begin{bmatrix} 1 \\ -a \\ \alpha_1 \\ -a\alpha_1 \\ \alpha_1^2 \\ \vdots \end{bmatrix}. \tag{4.14}$$

The dimension of the state space prediction (4.14) can be taken as large (or small) as needed to capture the desired function sequence.

The orthonormality expressed in (4.9) and (4.10) also exists in the time domain as

$$\sum_{n=0}^{\infty} k_{n,i} k_{n,j} = 1;\ i = j \tag{4.15}$$

$$\sum_{n=0}^{\infty} k_{n,i} k_{n,j} = 0;\ i \neq j \tag{4.16}$$

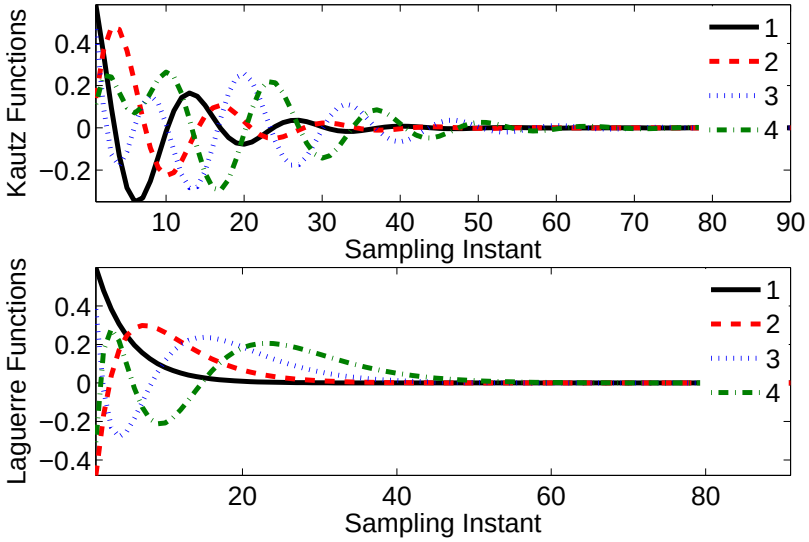

Figure 4.1: Coefficient of four Kautz and Laguerre functions

4.3.4 Special cases

Laguerre function as a special case when $(a, b) = (p, p)$

A Kautz function with $a = p$, $b = p$ where p is real, gives

$$\mathcal{K}_i = L_i$$

which demonstrate that Laguerre functions are a special case of Kautz functions. Clearly therefore, Kautz functions have more flexibility in terms of dominant dynamics than Laguerre functions and it is this flexibility that is explored in this chapter.

For completeness and to improve insight, Figure 4.1, shows the coefficients of the first four Kautz (with poles at $0.8 \pm 0.4j$) and Laguerre functions ($p = 0.8$). In both cases the speed of convergence is linked with the poles. In Figure 4.1 the convergence of Kautz polynomials are slower than that of Laguerre functions due to the poles being closer to the unit circle, but the oscillatory behaviour still allows the capture of some faster dynamics; hence it is expected that Kautz functions to offer more functionality than Laguerre functions by both facilitating slower convergence (to enlarge the region of attraction) in conjunction with more rapid transitions if required in near transients to improve performance.

Special case when $(a, b) = (0, 0)$

When $a = 0, b = 0$, the A_K matrix in (4.14) becomes

$$A_K = \begin{bmatrix} 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{4.17}$$

and the initial vector becomes

$$\mathcal{K}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots \end{bmatrix}^T. \tag{4.18}$$

Then, Kautz function become standard basis matrix with a set of pulses when $(a, b) = (0, 0)$. This property of Kautz is important because the previous work in the design of predictive control essentially uses this type of description for the incremental control trajectory, thus the MPC design using Kautz functions with $(a, b) = (0, 0)$, becomes equivalent to the traditional approach but it is also true for Laguerre function.

4.4 Kautz functions as an alternative parameterisation within MPC

Conventional algorithms use d.o.f. (or perturbation c_k) signals that have an impact on the region of attraction. Essentially adding n_c or d.o.f. expands the region of attraction to one where the state is able to enter the MAS with $\underset{\rightarrow}{c}_k = 0$ in at most n_c steps; such an expansion may therefore be smaller in conventional algorithms. The region of attraction is dominated by the choice of feedback gain K in terminal mode. This limitation is overcome in GERPC or optimisation dynamics MPC [25] which shows that one can use a highly tuned K and never the less obtained the region of attraction which is as large as can be obtained by using any detuned feedback gain. Despite significantly increasing the maximal stabilisable set, this formulation is based on ellipsoidal sets and needs $n_c \geq n_x$.

Laguerre functions have been proposed as a special case of GERPC to simplify the tradeoff between computational burden, region of attraction and performance [26, 202]. Laguerre functions with $p > 0$ evolve over an infinite horizon and the speed of convergence is linked directly to the time constant 'p'. When a small number of simple input

perturbations are not sufficient to regain feasibility using OMPC, Laguerre functions [26] provide an alternative trajectory for improving the feasibility with the same number of d.o.f..

As Laguerre functions are a special case of Kautz functions (i.e. for real p, $a = p$, $b = p$). Clearly therefore, Kautz functions have more flexibility in terms of parameter dynamics than Laguerre functions and it is an alternative parameterisation of the d.o.f. in order to further increase the region of attraction of OMPC algorithms. This section explores Kautz functions as an alternative parameterisation within OMPC algorithm.

4.4.1 Using Kautz functions within OMPC

A fundamental weakness of the OMPC algorithm is infeasibility when n_c steps are insufficient to move the initial state into the MAS. This weakness can be overcome by increasing the d.o.f. to allow more steps for reaching the MAS, but obviously at the expense of an increased computational burden. Another way of increasing the region of attraction is by detuning the terminal mode which may compromise performance. However, an alternative highlighted in [26] is to parameterise the d.o.f. differently so that the impact on the input predictions is over a longer horizon, thus relaxing the time requirement for entering the MAS. This section derives an algorithm which uses Kautz functions for this parameterisation, whereas the next section will compare these with the earlier Laguerre based approach.

There are two poles (real or complex) 'a' and 'b' that define the time scale for the input predictions using a combination of Kautz functions. The algorithm associated using Kautz functions is denoted as KOMPC for Kautz OMPC. Kautz functions can easily be used to redesign DMC/GPC achieving good performance and enlarge region of attraction, but here the focus is on dual-mode algorithms with guaranteed stability.

4.4.2 Kautz OMPC or KOMPC

Kautz functions are used to parameterise the perturbations c_k around the unconstrained optimal. The prediction using decision variables used in OMPC and KOMPC are put

side by side ($\underset{\rightarrow}{\gamma}_k$ denotes the KOMPC d.o.f.)

$$\underset{\rightarrow}{c}_k = \underbrace{\begin{pmatrix} c_k \\ \vdots \\ c_{k+n_c-1} \\ \vdots \end{pmatrix}}_{\text{OMPC}} \; or \; \underset{\rightarrow}{c}_k = \begin{pmatrix} \mathcal{K}_0^T \\ \vdots \\ \mathcal{K}_{n_c-1}^T \\ \vdots \end{pmatrix} \begin{pmatrix} \gamma_k \\ \vdots \\ \gamma_{k+n_c-1} \\ \vdots \end{pmatrix} = \underbrace{H_K \underset{\rightarrow}{\gamma}_k}_{\text{KOMPC}} \tag{4.19}$$

The key difference here from OMPC is that the H_K matrix has large number (in fact infinite) of rows. The number of rows of H_K can be truncated using the number of samples require to retain the steady state of Kautz functions (e.g. as shown in Figure 4.1). The performance index $J_{c,k}$ can be computed in terms of perturbation c_k as

$$J_{c,k} = \sum_{i=0}^{\infty} c_{k+i}^T S c_{k+i}. \tag{4.20}$$

However, from equation (4.19) note that $c_{k+i} = \mathcal{K}_i^T \underset{\rightarrow}{\gamma}_k$ and from equation (4.13) the new performance index becomes

$$\begin{aligned} J_{K,k} &= \sum_{i=0}^{\infty} \underset{\rightarrow}{\gamma}_k^T \mathcal{K}_i S \mathcal{K}_i^T \underset{\rightarrow}{\gamma}_k \\ &= \underset{\rightarrow}{\gamma}_k^T \left[\sum_{i=0}^{\infty} A_K^i \mathcal{K}_0 S \mathcal{K}_0^T (A_K^i)^T\right] \underset{\rightarrow}{\gamma}_k \end{aligned} \tag{4.21}$$

The MCAS can also be rewritten in the form

$$\mathcal{X}_c = \{x_k \in \mathbb{R}^{n_x} | \exists \underset{\rightarrow}{\gamma}_k \in \mathbb{R}^{n_c n_u}, M_K x_k + N_K H_K \underset{\rightarrow}{\gamma}_k \leq d_K\}. \tag{4.22}$$

for suitable M_K, N_K, H_K and d_K prediction matrices.

Algorithm 4.1. *Kautz OMPC (KOMPC)*

Off-line

1. *Determine the predicted cost, in terms of perturbations c_k is*

$$J_{c,k} = \sum_{i=0}^{\infty} c_{k+i}^T S c_{k+i}. \tag{4.23}$$

Substitute in from (4.13) *and* (4.19) *the KOMPC predictions of $c_{k+i} = \mathcal{K}_i^T \underset{\rightarrow}{\gamma}_k$ to*

give

$$J_{K,k} = \sum_{i=0}^{\infty} \underset{\rightarrow}{\gamma}_k^T \mathcal{K}_i S \mathcal{K}_i^T \underset{\rightarrow}{\gamma}_k. \tag{4.24}$$

Finally, substitute $\mathcal{K}_i = A_K \mathcal{K}_{i-1}$ *and hence*

$$J_{K,k} = \underset{\rightarrow}{\gamma}_k^T \left[\sum_{i=0}^{\infty} A_K^i \mathcal{K}_0 S \mathcal{K}_0^T (A_K^i)^T \right] \underset{\rightarrow}{\gamma}_k = \underset{\rightarrow}{\gamma}_k^T S_K \underset{\rightarrow}{\gamma}_k. \tag{4.25}$$

2. *Define the constraint inequalities associated to* (4.3) *in the form*

$$M_K x_k + N_K H_K \underset{\rightarrow}{\gamma}_k \leq d_K. \tag{4.26}$$

<u>*On-line*</u>

1. *At each sampling instant, perform the optimisation:*

$$\begin{aligned} &\underset{\rightarrow}{\gamma}_k^* = arg \min_{\underset{\rightarrow}{\gamma}_k} \; J_{K,k} \\ &s.t. \; M_K x_k + N_K H_K \underset{\rightarrow}{\gamma}_k \leq d_K. \end{aligned} \tag{4.27}$$

2. *Define* $\underset{\rightarrow}{c}_k = [\mathcal{K}_0^T, \ldots, \mathcal{K}_{n_c-1}^T] \underset{\rightarrow}{\gamma}_k^*$.

3. *Implement the first component of* $\underset{\rightarrow}{c}_k$*, that is* c_k *in the control law of* (4.2).

4. *If unconstrained control law is satisfying the constraints (i.e.* $x_k \in \mathcal{X}_0$*), the optimising* $\underset{\rightarrow}{c}_k$ *is zero so the control law is* $u_k = -Kx_k$.

Theorem 4.1. *If the values of* c_k *are restricted by* (4.19)*, then nevertheless, it is always possible to choose* $c_{k+i+1|k+1} = c_{k+i|k}$ $\forall i > 0$.

Proof. Define the predictions at two consequent samples as follows

$$\begin{bmatrix} c_{k|k} \\ c_{k+1|k} \\ c_{k+2|k} \\ \vdots \\ c_{k+n|k} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathcal{K}_0^T \\ \mathcal{K}_0^T (A_K)^T \\ \mathcal{K}_0^T (A_K^2)^T \\ \vdots \\ \mathcal{K}_0^T (A_K^n)^T \\ \vdots \end{bmatrix} \underset{\rightarrow}{\gamma}_k; \quad \begin{bmatrix} \underline{c_{k|k}} \\ c_{k+1|k+1} \\ c_{k+2|k+1} \\ \vdots \\ c_{k+n|k+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \underline{past} \\ \mathcal{K}_0^T \\ \mathcal{K}_0^T (A_K)^T \\ \vdots \\ \mathcal{K}_0^T (A_K^{n-1})^T \\ \vdots \end{bmatrix} \underset{\rightarrow}{\gamma}_{k+1}; \tag{4.28}$$

in order to make $c_{k+i|k+1} = c_{k+i|k}\ \forall i > 0$, it is sufficient to make

$$\mathcal{K}_0^T (A_K^i)^T \underset{\rightarrow}{\gamma}_k = \mathcal{K}_0^T (A_K^{i-1})^T \underset{\rightarrow}{\gamma}_{k+1}. \tag{4.29}$$

This is easily done by choosing $\underset{\rightarrow}{\gamma}_{k+1} = A_K \underset{\rightarrow}{\gamma}_k$. □

Theorem 4.2. *KOMPC has a guarantee of stability and recursive feasibility, in the nominal case.*

Proof. From Theorem 4.1, it is known that at time $k+1$, a shifted version of the optimal offset sequence c_k can be found if $\underset{\rightarrow}{\gamma}_{k+1} = A_K \underset{\rightarrow}{\gamma}_k$. So, substituting $\underset{\rightarrow}{\gamma}_k$ and $\underset{\rightarrow}{\gamma}_{k+1} = A_K \underset{\rightarrow}{\gamma}_k$ for time-steps k and $k+1$ respectively, in the cost function (4.21). At time $k+1$ the cost function will be less than that at time $k\ \forall x_0 \neq 0$, i.e.

$$J_{K,k+1} - J_{K,k} \leq - \underset{\rightarrow}{\gamma}_k^T \left[\mathcal{K}_0 S \mathcal{K}_0^T\right] \underset{\rightarrow}{\gamma}_k \leq 0.$$

Since $J_{K,k}$ is bounded below and $J_{K,k+1} \leq J_{K,k}$:

$$\implies J_{K,k+1} - J_{K,k} \to 0; \;\; k \to \infty,$$

therefore this implies that $J_{K,k}$ is Lyapunov function. Finally, note that for all states inside the MAS, the unconstrained optimal control law $u_k = -Kx_k$ will feasible, i.e. $c_k = \mathcal{K}_0^T \underset{\rightarrow}{\gamma}_k$ is repeatedly zeros. Hence, the Lyapunov stability of the origin follows from the fact that the MAS contains the origin in its interior. Recursive feasibility uses the same arguments. □

4.5 Numerical Examples

This section will illustrate the efficacy of the proposed KOMPC algorithm in comparison with LOMPC and OMPC using numerical examples. The main focus is to compare the trade off between feasibility gain and closed loop performance. The results will be presented in a way that is suitable for any number of state dimensions.

The aim is to compare two aspects

- The closed-loop performance for a range of initial conditions or, when feasible, how does the performance of a given algorithm compare to the global optimum.
- The region of attraction.

The global optimum J_{opt} is computed using OMPC with high n_c ($n_c = 20$ is used for numerical examples), it is used as a measure of how far the algorithms are from optimal.

4.5.1 Explanations of feasibility and performance comparison

This section used a very simple way of displaying relevant feasibility and performance information that does extend to arbitrary dimensions, does not require the computations burden.

The closed loop performance is measured by computing the performance index J_k in (3.27) over the time span where the system converges. The optimal performance index J_{opt} is computed using OMPC with high $n_c = 20$. The plots in Figure 4.4, 4.7, 4.9 and 4.11 show the normalised performance index for comparison. The regions of attraction in general are difficult to compare visually when larger than 2D plots. The region of attraction is computed by selecting different state directions and computing, how far out in these directions a feasible solution exists. The maximum distance point for various directions is denoted by σ (i.e. the distance from the origin to the boundary of MCAS), the various algorithms are then tested for $x_0 = \lambda\sigma$ ($\forall\lambda,\ 0 \leq \lambda \leq 1$). Clearly the larger the λ for which they are feasible, the larger the region of attraction in that specific direction, hereafter denoted as radius. Infeasibility is denoted by a zero in the normalised performance index plots.

The regions of attraction are also compared in Figure 4.3, 4.6, 4.8 and 4.10 using normalised radii obtained with OMPC using $n_c = 20$ as a function of n_c for OMPC, LOMPC and KOMPC.

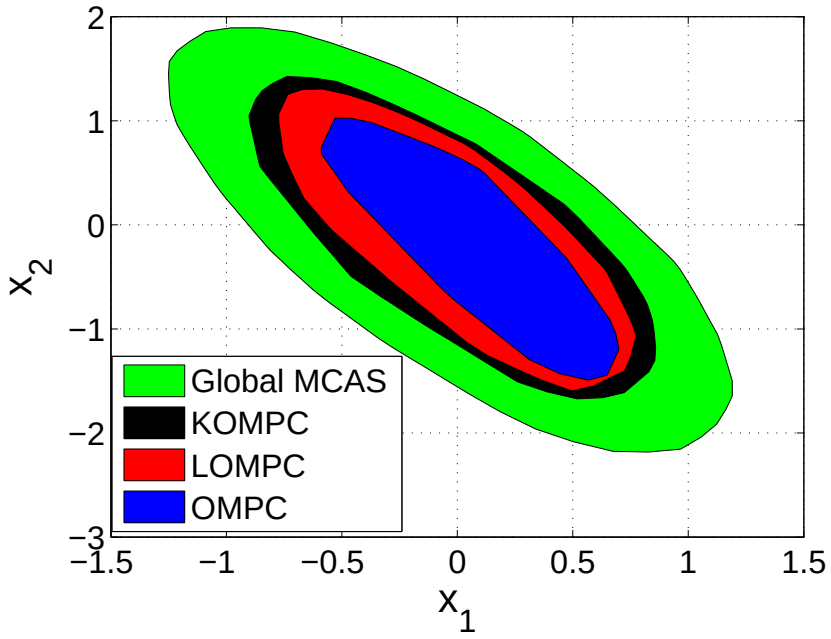

Figure 4.2: Comparison of MCAS for $n_c = 2$ for KOMPC,LOMPC,OMPC algorithms

4.5.2 Example 1 - $x \in \mathbb{R}^2$

The discrete-time state-space model and constraints are

$$A = \begin{bmatrix} 0.6 & -0.4 \\ 1 & 1.4 \end{bmatrix}; \; B = \begin{bmatrix} 0.2 \\ 0.05 \end{bmatrix}; \qquad C = \begin{bmatrix} 1 & -2.2 \end{bmatrix}; \tag{4.30}$$

$$\overline{\Delta u} = 0.4 = -\underline{\Delta u}; \; \overline{u} = 0.8; \; \underline{u} = -1.5; \; \overline{x} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} = -\underline{x}$$

The tuning parameters are $Q = I_{2\times 2}$, $R = 2$, $n_c = 2$, $p = 0.8$, $a = 0.8 + 0.45j$, $b = 0.8 - 0.45j$ and 116 state directions are chosen for the initial states. The pole locations of Laguerre and Kautz function dynamics are using the poles of closed loop system.

Figure 4.2 shows the region of attraction from which it is clear that KOMPC has a larger MCAS than both LOMPC and OMPC for the same number of d.o.f. i.e. $n_c = 2$. Figure 4.3 shows that the average MCAS radii as a function of n_c. For $n_c = 3$, KOMPC reach 85 % of MCAS for Global MPC. The plots of normalised cost against λ for a number of different state directions are plotted in Figure 4.4 for OMPC, LOMPC and KOMPC.

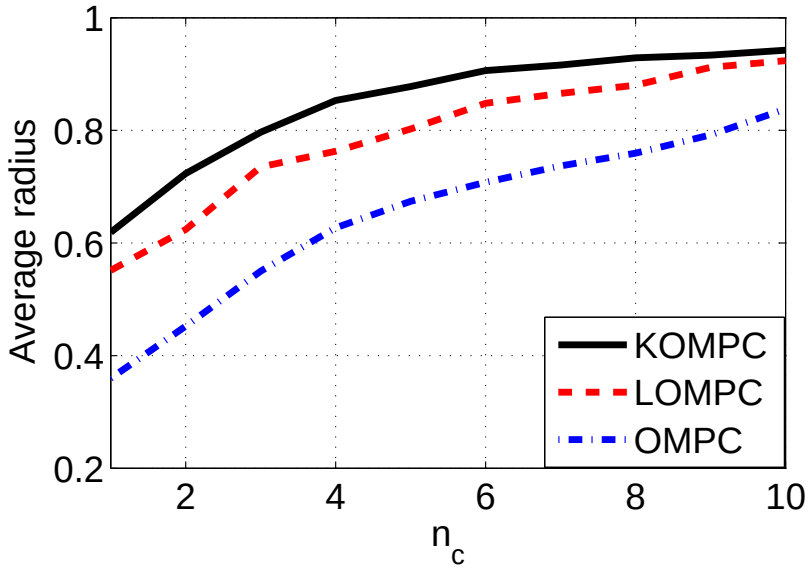

Figure 4.3: Comparison of normalised distance to global MCAS (i.e. OMPC with $n_c = 20$) as a function of n_c for OMPC, LOMPC and KOMPC.

The global optimal is computed with OMPC using $n_c = 20$ (i.e. Global MPC).

OMPC gives the global optimum performance for states well within the MCAS, but feasibility is severely restricted as the plots drops to zeros for small λ.

LOMPC enlarges the region of attraction as compared with OMPC, but a detriment to performance compared to the global optimum near its own MCAS boundary.

KOMPC enlarges the region of attraction as compared with LOMPC, but a detriment to performance compared to the global optimal near its own MCAS boundary. KOMPC has improved performance as compared with LOMPC near its own MCAS boundary.

4.5.3 Example 2 - $x \in \mathbb{R}^2$

The discrete-time state-space model and constraints are

$$A = \begin{bmatrix} 0.90133 & -0.1426 \\ 0.04752 & 0.9964 \end{bmatrix}; \; B = \begin{bmatrix} 0.2752 & 0.6243 \\ 0.1121 & 0.9471 \end{bmatrix}; \quad C = \begin{bmatrix} 0.4543 & 0.5623 \\ 0.0776 & 0.4545 \end{bmatrix}; \qquad (4.31)$$

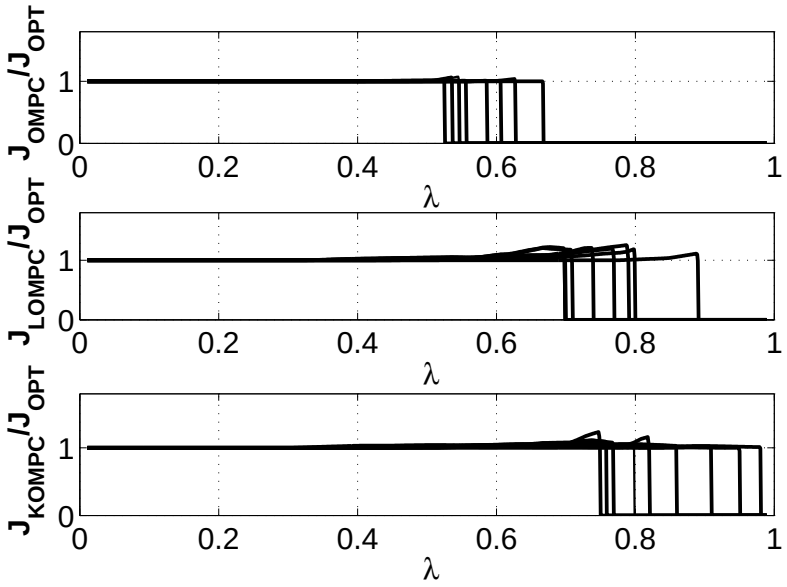

Figure 4.4: Normalised performance index $J_{OMPC}/J_{OPT}, J_{LOMC}/J_{OPT}, J_{KOMPC}/J_{OPT}$ for various state directions.

$$\overline{\Delta u} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -\underline{\Delta u};\ \overline{u} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}; = -\underline{u};\ \overline{x} = \begin{bmatrix} 20 \\ 15 \end{bmatrix} = -\underline{x}$$

The tuning parameters are $Q = C^T C$, $R = I$, $n_c = 2$, $p = 0.5$, $a = 0.6$. $b = 0.7$ and 196 state directions are chosen for the initial states.

Figure 4.5, 4.6 and 4.7 show the feasibility/performance results for OMPC, LOMPC and KOMPC algorithms. It is clear that KOMPC has a larger MCAS than both LOMPC and OMPC for the same number of d.o.f. i.e. $n_c = 2$. Figure 4.6 shows that for $n_c = 2$, KOMPC reaches 99% and LOMPC reaches 90% of MCAS for Global MPC. For $n_c = 3$, both KOMPC and LOMPC reach, to within less than 1%, the MCAS for Global MPC. Figure 4.7 shows that:

OMPC gives improved performance for states well within the MCAS as comparison with LOMPC and KOMPC, but feasibility is severely restricted as the plots drops to zeros for small λ.

LOMPC enlarges the region of attraction as compared with OMPC, but a detriment to performance compared to the global optimum.

KOMPC enlarges the region of attraction and gives practically the same performance as the global optimum.

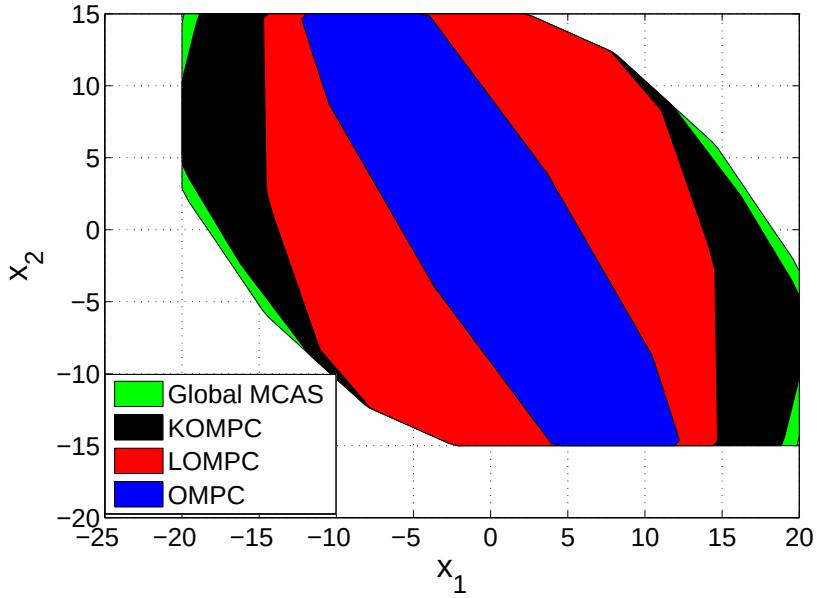

Figure 4.5: Comparison of MCAS for $n_c = 2$ for KOMPC,LOMPC,OMPC algorithms

4.5.4 Example 3 - $x \in \mathbb{R}^3$

For this example the discrete-time state-space model and constraints are given by

$$A = \begin{bmatrix} 1.4000 & -0.1050 & -0.1080 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \; B = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix}; \\ C = \begin{bmatrix} 5 & 7.5 & 0.5 \end{bmatrix}. \tag{4.32}$$

$$\overline{\Delta u} = 0.02 = -\underline{\Delta u}; \; \overline{u} = 0.04 = -\underline{u}; \; \overline{y} = 1.2 = -\underline{y}.$$

The tuning parameters are $Q = C^T C$, $R = 2$, $n_c = 2$, $p = 0.8$, $a = 0.8 + 0.26j$, $b = 0.8 - 0.26j$ and 296 state directions are chosen for the initial states. Figure 4.8 and 4.9 show the feasibility/performance results for OMPC, LOMPC and KOMPC algorithms. Figure 4.8 shows that for $n_c = 3$, KOMPC reaches 100% of MCAS for Global MPC, whereas LOMPC requires 4 d.o.f.. Figure 4.9 shows that:

OMPC has improved performance but very limited feasibility.

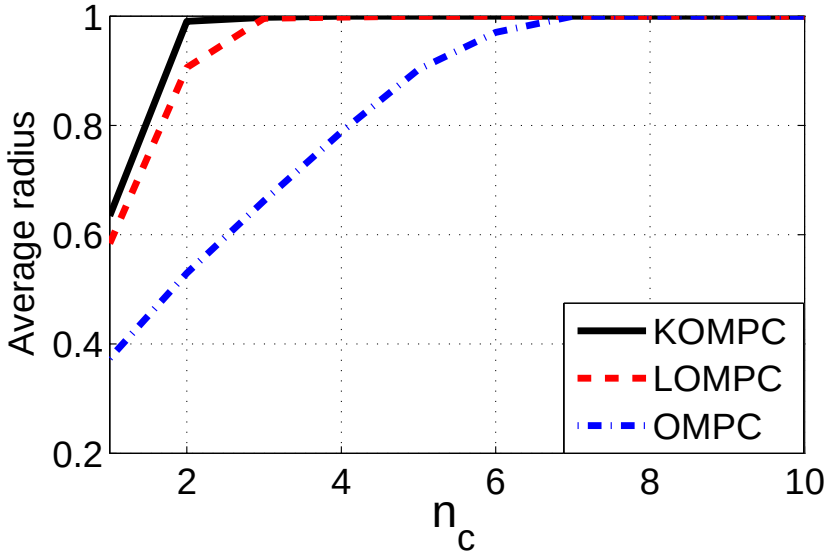

Figure 4.6: Comparison of normalised distance to global MCAS (i.e. OMPC with $n_c = 20$) as a function of n_c for OMPC, LOMPC and KOMPC.

LOMPC has noticeably enlarge the region of attraction than OMPC, but with a small performance loss in the mid ranges of λ.

KOMPC has improved region of attraction and performance than LOMPC, but a detrimental to performance in the mid ranges of λ.

4.5.5 Example 4 - $x \in \mathbb{R}^4$

For this example the discrete-time state-space model and constraints are

$$A = \begin{bmatrix} 0.9146 & 0 & 0.0405 & 0.1 \\ 0.1665 & 0.1353 & 0.0058 & -0.2 \\ 0 & 0 & 0.1353 & 0.5 \\ -0.2 & 0 & 0 & 0.8 \end{bmatrix}; \qquad B = \begin{bmatrix} 0.0544 & -0.0757 \\ 0.0053 & 0.1477 \\ 0.8647 & 0 \\ 0.5 & 0.2 \end{bmatrix};$$
$$C = \begin{bmatrix} 1.7993 & 13.2160 & 0 & 0.1 \\ 0.8233 & 0 & 0 & -0.3 \end{bmatrix}. \tag{4.33}$$

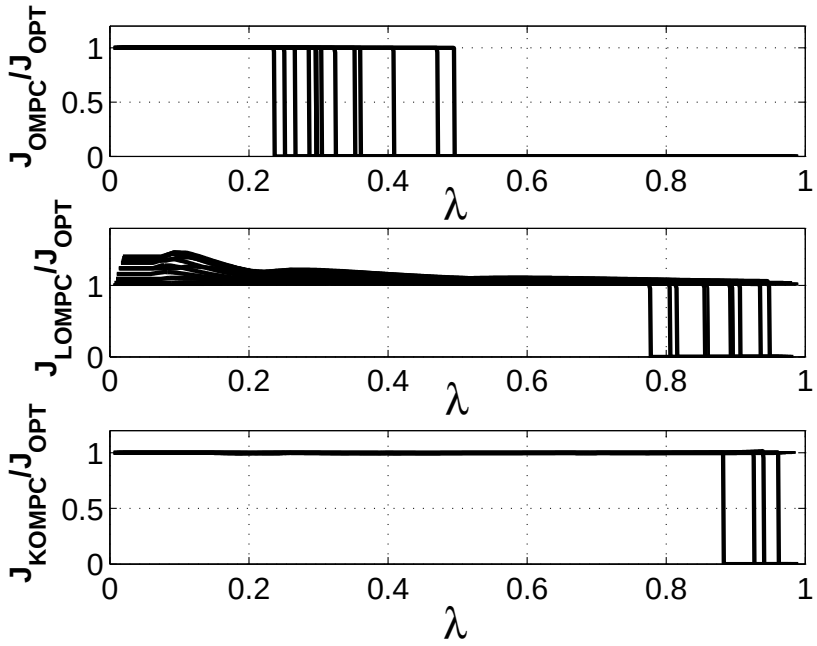

Figure 4.7: Normalised performance index $J_{OMPC}/J_{OPT}, J_{LOMC}/J_{OPT}, J_{KOMPC}/J_{OPT}$ for various state directions.

$$\overline{\Delta u} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = -\underline{\Delta u};\ \overline{u} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} = -\underline{u};\ \overline{x} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} = -\underline{x}.$$

The tuning parameters are $Q = I_{4\times4}$, $R = I_{2\times2}$, $n_c = 2$, $p = 0.8$, $a = 0.5$, $b = 0.8$ and 1040 state directions are chosen for the initial states. The pole locations of Laguerre and Kautz function dynamics are using the poles of closed loop system.

Figure 4.10 and 4.11 show the comparison results of performance/feasibility for OMPC, LOMPC and KOMPC algorithms for the chosen state directions. Figure 4.10 shows that, KOMPC gets to within 100% of the global MCAS with just 3 d.o.f. whereas, LOMPC requires 4 d.o.f.. Figure 4.10 shows that KOMPC is detrimental to performance but less in comparison with LOMPC.

4.5.6 Regions of attraction

The numerical examples show that the Kautz function parameterisation is more flexible than previously proposed Laguerre function as a means of enlarging the region of attrac-

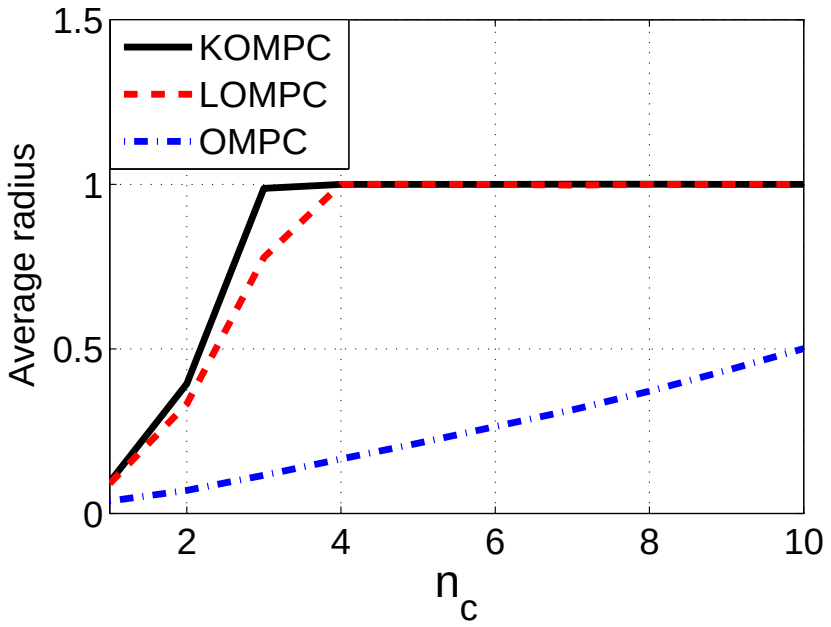

Figure 4.8: Comparison of normalised distance to global MCAS (i.e. OMPC with $n_c = 20$) as a function of n_c for OMPC, LOMPC and KOMPC.

tion when the number of d.o.f. are limited and with little detriment to the closed loop performance. There is an anticipated benefit to feasibility because if the initial state is far away from the MAS, then n_c steps (n_c small) will be insufficient and thus OMPC becomes infeasible. The restriction in feasibility may be regained using the Kautz function parameterisations.

For completeness, Table 4.1 shows the volume of MCAS for the numerical examples presented in this section. The MCAS volume is calculated using the Multi-Parametric Toolbox (MPT) [123]. This data is an objective measure of MCAS and the observation is that the KOMPC may enlarge the region of attraction by utilising the d.o.f. more effectively.

4.5.7 Closed loop performance and computational load

Performance comparisons will be based on computing performance ratio (i.e. J_{OMPC}/J_{opt}, J_{LOMPC}/J_{opt} and J_{KOMPC}/J_{opt}) over the closed loop responses. The lower the performance ratio i.e. $J_{KOMPC}/J_{opt} \leq 1$, the better the performance. The comparison of all numerical examples is plotted in Figure 4.4, 4.7, 4.9 and 4.11 with different initial conditions. It is observed that the KOMPC may improve the feasibility with little or no

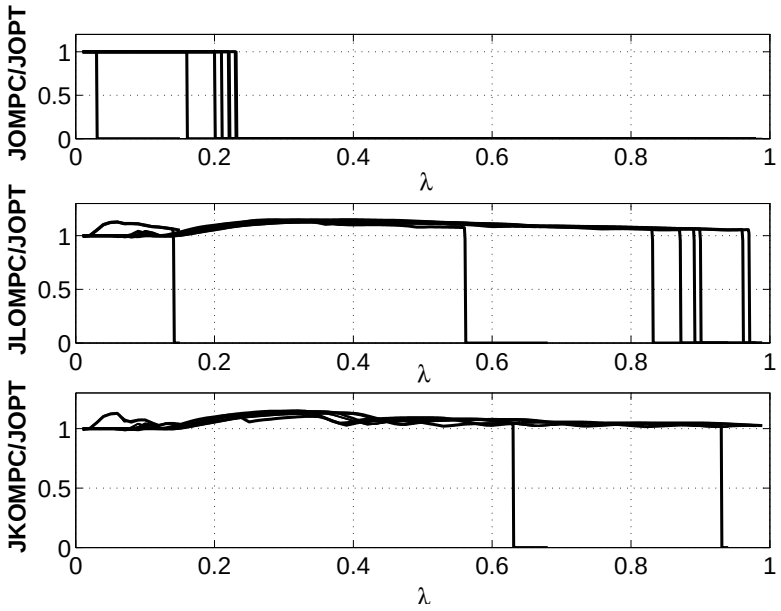

Figure 4.9: Normalised performance index $J_{OMPC}/J_{OPT}, J_{LOMC}/J_{OPT}, J_{KOMPC}/J_{OPT}$ for various state directions.

Table 4.1: Volume comparison for $n_c = 2$

	OMPC	LOMPC	KOMPC
Example 1	1.3841	2.2723	2.8871
Example 2	174.8301	745.9069	939.7816
Example 3	0.0526	0.2712	0.2945
Example 4	71.6438	73.1401	73.3729

detriment to close loop performance.

The computational load comparison will be based solely on the number of d.o.fs.. The d.o.f is similar for all the algorithms, thus KOMPC has the same computational load as compared with OMPC and LOMPC.

On the other hand, KOMPC provides two tuning parameters i.e. 'a' and 'b' for the size of the region of attraction and closed loop performance. Thus one could even envisage changing these parameters to improve the trade off between region of attraction/performance where required and thus deploying no extra d.o.f..

Remark 4.5.1. The parameterised matrix H_K in (4.19) is generated using (4.8). It is easy to use network realisation for complex poles.

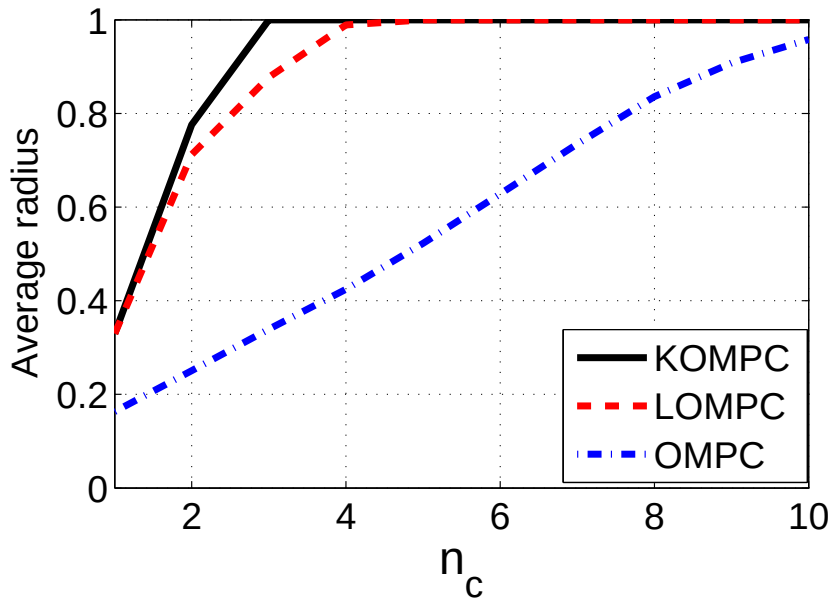

Figure 4.10: Comparison of normalised distance to global MCAS (i.e. OMPC with $n_c = 20$) as a function of n_c for OMPC, LOMPC and KOMPC.

4.6 Conclusion

The chapter has argued for the potential benefits of Kautz functions as more flexible parameterisation for maximising the region of attraction in conventional MPC algorithms with a fixed number of d.o.f.. It has been shown through examples that the region of attraction can be maximised without too much degrading the closed loop performance. It has also been shown that a simple re-parameterisation of the degrees of freedom within the input predictions can achieve improved performance and enlarge MCAS. However, of more significance, the chapter has tackled the question concerning the earlier proposed use of Laguerre functions to parameterise the d.o.f. in the predictions and clearly demonstrated that obvious alternatives do exist and in fact, this chapter indicates that Kautz functions may indeed be preferable to Laguerre in general.

This chapter focussed on just one possible parameterisation and coming chapters will tackle the question of whether there exists other more flexible alternative choices and a 'systematic' method for choosing the best parameterisation for any given problem. Moreover, it has been noted that computational burden is linked not only to the number of d.o.f. but also the optimal structure and thus the next step is to consider computational benefits using more flexible parameterisation.

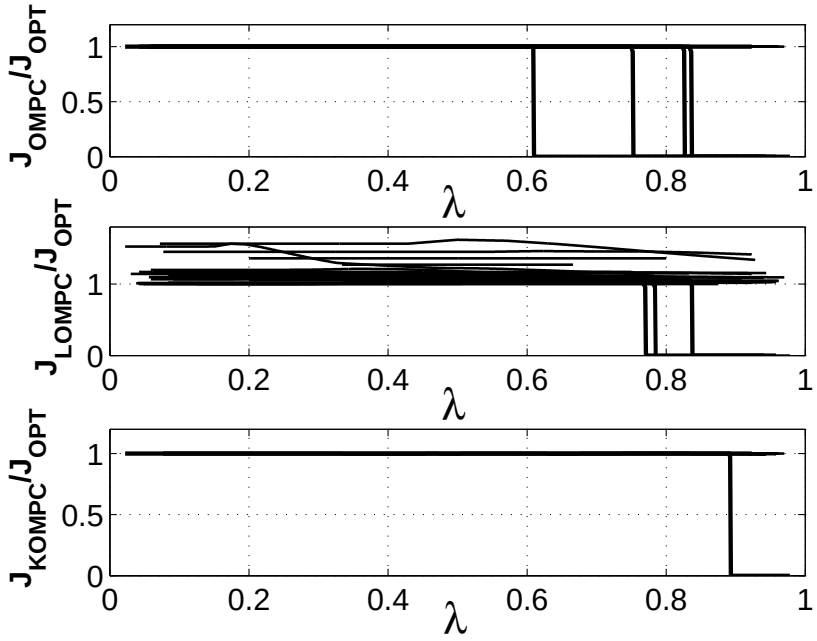

Figure 4.11: Normalised performance index $J_{OMPC}/J_{OPT}, J_{LOMC}/J_{OPT}, J_{KOMPC}/J_{OPT}$ for various state directions.

Chapter 5

Generalised parameterisation for optimal predictive control

This chapter presents an **original contribution** to the thesis. It generalises approaches to predictive control based on Laguerre and Kautz functions in a format to simplify the trade off between region of attraction, performance and inexpensive optimisation. It is shown that Laguerre and Kautz are special cases of generalised functions and thus one can give a more general parameterisation using higher order functions. Specifically, a simple but efficient algorithm that uses generalised functions to parameterise the degrees of freedom in an optimal predictive control is presented. The efficacy of the proposed parameterisation within existing predictive control algorithms that use a similar strategy including GERPC using [25], is demonstrated by examples. It is also shown that the propose algorithm has standard convergence and feasibility guarantees.

This chapter is organised as follows: Section 5.1 presents the introduction and motivation of the chapter; Section 5.2 presents the generalised functions and its properties; Section 5.3 presents the algorithms that uses generalised functions; Section 5.4 presents numerical examples; and finally Section 5.5 gives the conclusion of the chapter.

5.1 Introduction

Dual mode MPC allows simple guarantees for stability and recursive feasibility [11] and thus is a mechanism much favoured in academia. Dual mode MPC prescribes the use of

transient degrees of freedom (d.o.f.) followed by a terminal control law tuned to give high performance, however this will often result in a relatively small region of attraction [13]. One strategy to increase the volume of the region of attraction while retaining good performance is to increase the number of d.o.f.; of course this increases the computational burden which will have a limit. To overcome this limitation, [25] optimises prediction dynamics to enlarge the region of attraction using a highly tuned terminal control law based on ellipsoidal sets and the prediction horizon needs not exceed the system dimension. The proposal is based on ellipsoidal sets, which may result in conservative polytopic regions. This chapter proposes an alternative approach based on generalised function parameterisation to simplify the trade off between the region of attraction, performance and computational burden. The numerical results are compared with optimisation dynamics in [25].

Kautz functions in Section 4.3, have been proposed as an effective mean of parameterising the input predictions using Kautz functions. Specifically it was shown that such a parameterisation of the d.o.f. in dual mode MPC can enlarge the region of attraction without too much compromise to the performance and computational burden. Nevertheless, one key question was still left unanswered: is there a generalisation or indeed is there an alternative to Laguerre and Kautz functions which is better still? These questions are tackled in this chapter.

Laguerre and Kautz are 1st and 2nd order parameterisations and thus it is logical to consider whether higher order or generalised parameterisation techniques further improve the region of attraction while maintaining performance. Specifically, the intent is to explore the generalisation of the parameterisation using orthonormal basis function [2, 18,26,138,202] based approaches that have been developed recently in Section 4.3 and [26]. Hence, in line with the proposals in Section 4.3, higher order basis functions are tested as these are generalisation of Laguerre and Kautz functions. It is demonstrated that higher order orthonormal basis functions are an effective alternative to the standard basis set of Laguerre and Kautz functions for parameterising d.o.f. within an optimal MPC and moreover may be more effective and offer more variety in the key characteristics. The higher order orthonormal basis functions are also a special case of generalised prediction framework in [25] and deploying the particular lower triangular structure. The numerical details for generalisation using higher order basis functions are provided but the issue of how to make a systematic choice of 'function order' and parameter to best meet a specific objective is left for coming chapters.

5.2 Generalised Orthonormal Basis functions

Orthonormal basis functions [18,26] like Laguerre and Kautz functions have been popular in filtering, system identification and control design where a few parameters are enough to describe the behaviour of the system. The properties depend upon the selection of scaling factors and the complexity and accuracy of the description increases as the network dimension increases. In a similar way, Laguerre and Kautz functions can be used for the parameterisation of the d.o.f. in MPC to enlarge the region of attraction and improve performance. This section explores generalised basis functions for enlarging the region of attraction and without detriment to performance of an Optimal MPC algorithm.

5.2.1 Generalised higher Order Network

The higher order network is defined as follows

$$g_i(z) = \sqrt{(1-a_i^2)\dots(1-a_n^2)}\frac{(z^{-1}-a_1)^{i-1}\dots(z^{-1}-a_n)^{i-1}}{(1-a_1z^{-1})^{i}\dots(1-a_nz^{-1})^{i}} \qquad (5.1)$$
$$0 \le a_k < 1, \qquad k = 1,\dots,n,$$

where '$a_1,\dots,a_n$' are poles of the 'nth' order discrete time generalised network. The free parameters, '$a_1,\dots,a_n$' are selected by the user; these are also called the scaling factors. However, the inverse z-transform of the higher order networks do not lead to a compact expression of the orthonormal functions in the time-domain so, as in equation (5.1), a state-space type of representation is preferred and derived briefly here

$$g_i(z) = g_{i-1}(z)\frac{(z^{-1}-a_1)^{i-1}\dots(z^{-1}-a_n)^{i-1}}{(1-a_1z^{-1})^{i}\dots(1-a_nz^{-1})^{i}} \qquad (5.2)$$
$$0 \le a_k < 1, \qquad k = 1,\dots,n,$$

with $g_1(z) = \frac{\sqrt{(1-a_1^2)\dots(1-a_n^2)}}{(1-a_1z^{-1})\dots(1-a_nz^{-1})}$. The discrete time orthonormal functions are expressed in a vector form as

$$g_n = [g_{n,1}, g_{n,2}, \dots]^T. \qquad (5.3)$$

Taking advantages of the network realisation in equation (5.2), the set of discrete time orthonormal functions satisfies the following difference equation (see Section A.2 for

further details)

$$G_{k+1} = A_G G_k, \tag{5.4}$$

let the size of A_G is $N \times N$ and is a function of parameters '$a_1, \ldots, a_m$', where 'm' is order of dynamics and 'N' is the dimension of prediction matrix A_G.

State space representation of $4th$ order network

For example, in case of $m = 4$ with $N = 6$ state space is given by

$$\begin{bmatrix} g_{1,k+1} \\ g_{2,k+1} \\ g_{3,k+1} \\ g_{4,k+1} \\ g_{5,k+1} \\ g_{6,k+1} \end{bmatrix} = G_{k+1} = \underbrace{\begin{bmatrix} a_2 & 0 & 0 & 0 & 0 & 0 \\ a_2 & a_3 & 0 & 0 & 0 & 0 \\ a_2 & a_3 & a_4 & 0 & 0 & 0 \\ -a_1a_2 & -a_1a_3 & (1-a_1a_4) & a_1 & 0 & 0 \\ a_1a_2^2 & a_1a_2a_3 & -a_2(1-a_1a_4) & (1-a_1a_2) & a_2 & 0 \\ -a_1a_2^2a_3 & -a_1a_2a_2^2 & a_2a_3(1-a_1a_4) & -a_3(1-a_1a_2) & (1-a_1a_2) & a_3 \end{bmatrix}}_{A_G} G_k,$$

$$G_0 = \sqrt{(1-a_1^2)\ldots(1-a_4^2)} \begin{bmatrix} 1 & 1 & 1 & -a_1 & a_1a_2 & -a_1a_2a_3 \end{bmatrix}^T. \tag{5.5}$$

5.2.2 Orthonormality of generalised higher order functions

The orthonormality of generalised higher order functions may be proved in both frequency and time domain similarly as proved for Kautz functions. In the frequency domain, this orthonormality is expressed in terms of the orthonormal equations for $g_m(m = 1, 2, \ldots)$ from (5.1) as

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} g_m(e^{jw}) g_n(e^{jw})^* dw = 1; \;\; m = n \tag{5.6}$$

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} g_m(e^{jw}) g_n(e^{jw})^* dw = 0; \;\; m \neq n \tag{5.7}$$

where $(.)^*$ denotes the complex conjugate of $(.)$. Similarly in time domain, using network generalisation sequence from (5.3) as

$$\sum_{n=0}^{\infty} g_{n,i} g_{n,j} = 1; \; i = j \tag{5.8}$$

$$\sum_{n=0}^{\infty} g_{n,i} g_{n,j} = 0; \; i \neq j \tag{5.9}$$

5.2.3 Laguerre and Kautz function as a special case

A higher order orthonormal function with a_j, $\forall j = 1, \ldots, n$ gives

$$\begin{aligned} G_k &= L_k && if \; a_j = p, \\ G_k &= \mathcal{K}_k && if \; a_j = [a, b], \end{aligned} \tag{5.10}$$

which demonstrates that Laguerre and Kautz functions are a special case of higher order orthonormal functions. This may be proved using equation (5.1) and (5.5) similarly as done in previous chapter Section 4.3.4.

Clearly therefore, G_k has more flexibility in terms of dominant dynamics than Laguerre and Kautz functions and it is this flexibility that is explored in this chapter.

Remark 5.2.1. If generalised higher order network uses $a_j = 0$, then $G_0 = [1, 0, \ldots]$ and A_G becomes a shift matrix, that is ones on the lower diagonal. In this case generalised functions are equivalent to standard basis function.

5.3 Using generalised orthonormal functions in OMPC

The basic concept of dual mode MPC and indeed LOMPC are preserved where input perturbation c_k is parameterised in terms of generalised functions. Hence the input prediction perturbation using a generalise function is given by:

$$\underrightarrow{c}_k = \begin{pmatrix} c_k \\ c_{k+1} \\ \vdots \end{pmatrix} = \begin{pmatrix} G_0^T \\ G_1^T \\ \vdots \end{pmatrix} \underrightarrow{\rho}_k = H_G \underrightarrow{\rho}_k \tag{5.11}$$

where $\underset{\rightarrow}{\rho}_k$ is the n_G dimension decision variable when using the first n_G column of H_G. In fact, the only conceptual difference between Laguerre OMPC (LOMPC), Kautz OMPC (KOMPC) and generalised OMPC (GOMPC) is the definition of the H_G matrix but otherwise the conceptual steps and algebra is essentially the same; in consequence the presentation of GOMPC next is deliberately concise. The prediction cost using generalised function parameterisation in terms of $\underset{\rightarrow}{\rho}_k$ is given by

$$J_{G,k} = \underset{\rightarrow}{\rho}_k^T \left(\sum_{i=0}^{\infty} A_G^i G_0 S G_0^T {A_G^i}^T \right) \underset{\rightarrow}{\rho}_k = \underset{\rightarrow}{\rho}_k^T S_G \underset{\rightarrow}{\rho}_k \tag{5.12}$$

with $c_{k+i} = G_i^T \underset{\rightarrow}{\rho}_k$ and from $G_i = A_G G_{i-1}$. The MCAS is calculated in a similar manner to LOMPC, the main difference in the calculation is the use of transformation matrix H_G instead of H_L (more efficient/precise computations do exist but are not interesting enough to discuss here).

Algorithm 5.1. *Generalised OMPC (GOMPC)*

Off-line

1. *Select the order of prediction dynamics, that is the number of poles a_j in A_G.*

2. *Select specific values for the poles a_j.*

3. *Determine the predicted cost, in terms of perturbations c_k is*

$$J_{c,k} = \sum_{i=0}^{\infty} c_{k+i}^T S c_{k+i}. \tag{5.13}$$

Substitute in from (5.5) *and* (5.11) *the GOMPC predictions of $c_{k+i} = G_i^T \underset{\rightarrow}{\rho}_k$ to give*

$$J_{G,k} = \sum_{i=0}^{\infty} \underset{\rightarrow}{\rho}_k^T G_i S G_i^T \underset{\rightarrow}{\rho}_k. \tag{5.14}$$

Finally, substitute $G_i = A_G G_{i-1}$ and hence:

$$J_{G,k} = \underset{\rightarrow}{\rho}_k^T \left[\textstyle\sum_{i=0}^{\infty} A_G^i G_0 S G_0^T (A_G^i)^T \right] \underset{\rightarrow}{\rho}_k = \underset{\rightarrow}{\rho}_k^T S_G \underset{\rightarrow}{\rho}_k. \tag{5.15}$$

4. *Define the constraint inequalities associated to* (4.20) *in the form*

$$M_G x_k + N_G H_G \underset{\rightarrow}{\rho}_k \leq d_G, \tag{5.16}$$

for suitable M_G, N_G *and* d_G *prediction matrices.*

<u>*On-line*</u>

1. *At each sampling instant, perform the optimisation:*

$$\begin{aligned} &\underset{\rightarrow}{\rho}{}^*_k = arg \min_{\underset{\rightarrow}{\rho}_k} J_{G,k} \\ &s.t.\ M_G x_k + N_G H_G \underset{\rightarrow}{\rho}_k \leq d_G. \end{aligned} \tag{5.17}$$

2. *Define* $\underset{\rightarrow}{c}_k = [G_0^T, \ldots, G_{n_c-1}^T] \underset{\rightarrow}{\rho}{}^*_k$.

3. *Implement the first component of* $\underset{\rightarrow}{c}_k$, *that is* c_k *in the control law of* (4.2).

4. *If unconstrained control law is satisfying the constraints (i.e.* $x_k \in \mathcal{X}_0$*), the optimising* $\underset{\rightarrow}{c}_k$ *is zero so the control law is* $u_k = -Kx_k$.

Theorem 5.1. *The GOMPC algorithm has a guarantee of stability and recursive feasibility in the nominal case.*

Proof. The essence of the proof is to show that the cost function $J_{G,k}$ is monotonically decreasing. First, assume that at sample k and $k+1$ the optimal sequence of prediction input perturbations are given as:

$$\underset{\rightarrow}{c}{}^T_k = [c_k^T, c_{k+1|k}^T, c_{k+2|k}^T, \cdots]; \quad \underset{\rightarrow}{c}{}^T_{k+1} = [c_{k+1|k+1}^T, c_{k+2|k+1}^T, \cdots]$$

Also note that $J_{G,k} = \sum_{i=0}^{\infty} c_{k+i}^T S c_{k+i}$. Consequently, if one were to choose that $c_{k+i|k+1} = c_{k+i|k}$ then

$$J_{G,k+1} = J_{G,k} - c_k^T S c_k$$

In other words, it is sufficient to show that <u>if one can choose</u> at each sampling instant the same input predictions as selected at the previous sample, then convergence follows;

it is well known in the literature that recursive feasibility in the nominal case also follows automatically from this [5, 6]. Hence the proof is confirmed if one can show that the choice $[0^T, \underset{\rightarrow}{c}_{k+1}^T] = \underset{\rightarrow}{c}_k^T$ is always possible. From the definitions in (5.11) this requires that there exists a $\underset{\rightarrow}{\rho}_{k+1}$ such that:

$$\begin{pmatrix} G_1^T \\ G_2^T \\ \vdots \end{pmatrix} \underset{\rightarrow}{\rho}_k = \begin{pmatrix} G_0^T \\ G_1^T \\ \vdots \end{pmatrix} \underset{\rightarrow}{\rho}_{k+1}; \tag{5.18}$$

However, it has already been established that $G_{k+1} = A_G G_k$ and thus (5.18) can be represented as:

$$\begin{pmatrix} G_0^T A_G^T \\ G_0^T [A_G^T]^2 \\ \vdots \end{pmatrix} \underset{\rightarrow}{\rho}_k = \begin{pmatrix} G_0^T \\ G_0^T [A_G]^T \\ \vdots \end{pmatrix} \underset{\rightarrow}{\rho}_{k+1}; \quad \Rightarrow \quad A_G^T \underset{\rightarrow}{\rho}_k = \underset{\rightarrow}{\rho}_{k+1} \tag{5.19}$$

which can be satisfied because A_G is full rank by definition. Finally, note that for all states inside the MAS, the unconstrained optimal control law $u_k = Kx_k$ will feasible i.e. $c_k = G_0^T \underset{\rightarrow}{\rho}_k$ is repeatedly zeros. Hence, the Lyapunov stability of the origin follows from the fact that the MAS contains the origin in its interior. Recursive feasibility uses the same arguments.

□

Remark 5.3.1. GOMPC gives the most flexibility in the shapes of the input predictions, but at the price of a more involved prediction structure. Where less flexibility is required, a designer may choose to use KOMPC, LOMPC or even OMPC. The key point is that this suite of parameterisations offers a systematic path to follow when for example, OMPC is not giving adequate feasibility for reasonable values of n_c.

Remark 5.3.2. For multivariable signals, one can easily modify the above algebra, this detail is omitted.

5.4 Numerical examples

This section will illustrate the efficacy of the proposed GOMPC algorithm in comparison with ERPC [21], GERPC [25], LOMPC, KOMPC and GOMPC using numerical examples for the linear time invariant case. The main focus of this section is a comparison based on performance, region of attraction and computational burden. The comparison is based on the highly tuned GERPC algorithm with suboptimal choices of LOMPC, KOMPC and GOMPC. The pole locations of parameterised dynamics are selected to be equal to or in the vicinity of pole(s) of the of the closed loop stable system, that is the one that arises from the terminal control law. All numerical examples are implemented using both symmetric and non-symmetric constraints. The region of attraction is calculated using Multi-Parametric Toolbox (MPT) [123].

5.4.1 Example 1 - $x \in \mathbb{R}^2$

For this example the discrete-time state-space model and constraints double integral used in [24], [25] is given as follows

$$A = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix};\ B = \begin{bmatrix} T_s^2 \\ T_s \end{bmatrix};\ C = \begin{bmatrix} 1 & 0 \end{bmatrix}; \tag{5.20}$$

with $T_s = 0.05$.

Example 1 (a)

$$\overline{u} = 1 = -\underline{u};\ \overline{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -\underline{x}.$$

Example 1 (b)

$$\overline{u} = 1;\ \underline{u} = -0.5;\ \overline{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} =\ \underline{x} = -\begin{bmatrix} 1 \\ 1.5 \end{bmatrix}.$$

The tuning parameters are $Q = C^T C, R = 1, n_c = 2, n_G = 2$, Laguerre parameter $p = 0.92$, Kautz parameters $(a, b) = (0.92, 0.7)$ and $\gamma = \infty$.

Table 5.1, Figure 5.1 and 5.2 show the region of attraction from which it is clear that KOMPC has a larger MCAS than LOMPC, GERPC and ERPC for the same number of d.o.f. i.e. $n_c = 2$. This increase comes at the price of an increase in the number of constraints in the online QP problem, as can be seen in the first and second row of Table

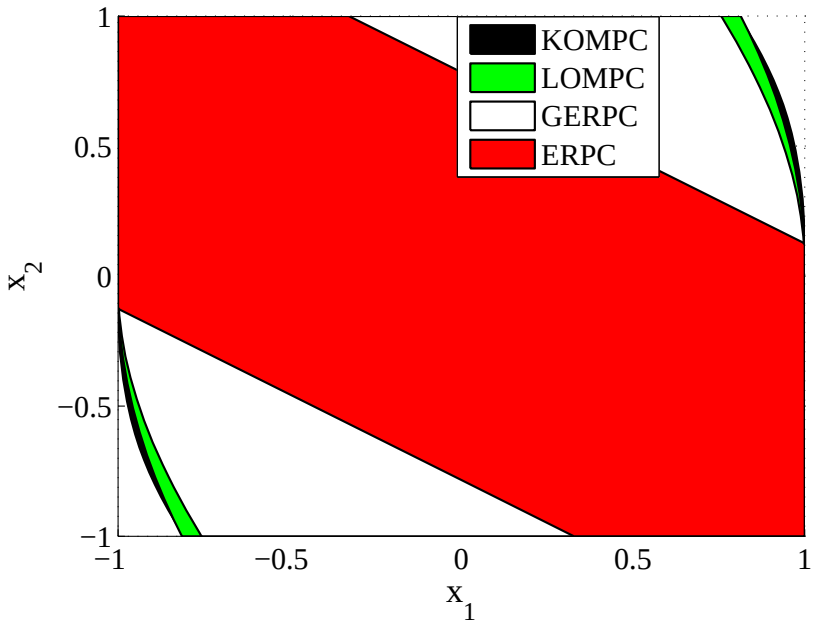

Figure 5.1: Comparison of MCAS for $n_c = n_G = 2$ for KOMPC, LOMPC, GERPC and ERPC algorithms for Example 1(a).

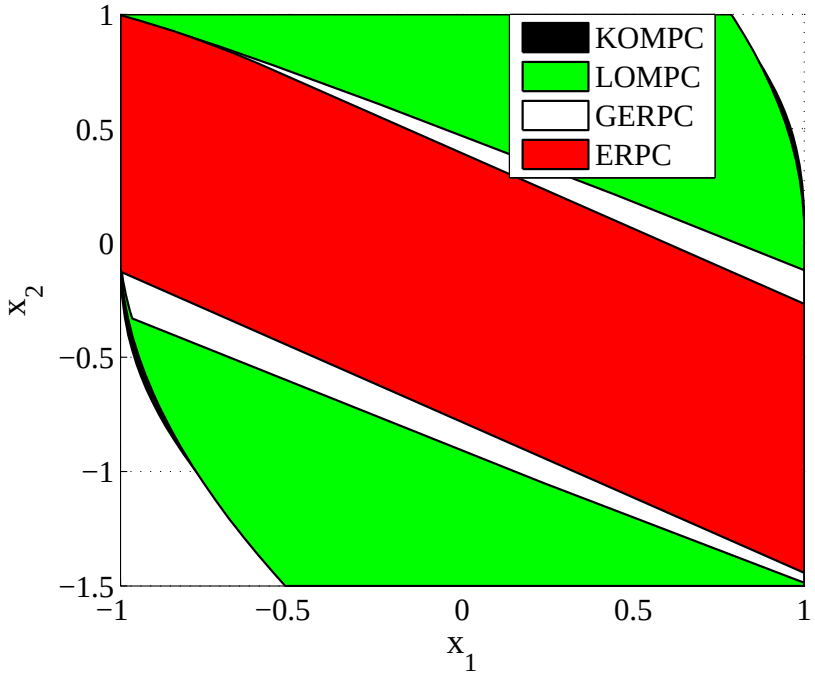

Figure 5.2: Comparison of MCAS for $n_c = n_G = 2$ for KOMPC, LOMPC, GERPC and ERPC algorithms for Example 1(b).

5.2. However, the number of inequalities for KOMPC and LOMPC are less than GERPC algorithm. The KOMPC is tuned using LOMPC parameter selection (i.e. $a = p$), so it is observed that there is an insignificant difference between the resulting region of attractions. For non-symmetric constraints in Example 1(b), the region of attractions for KOMPC and LOMPC are significantly enlarged compared to GERPC.

5.4.2 Example 2 - $x \in \mathbb{R}^2$

For this example the discrete-time state-space model and constraints are

$$A = \begin{bmatrix} 0.6 & -0.4 \\ 1 & 1.4 \end{bmatrix};\ B = \begin{bmatrix} 0.2 \\ 0.05 \end{bmatrix};\ C = \begin{bmatrix} 1 & -2.2 \end{bmatrix}; \tag{5.21}$$

Example 2 (a)

$$\overline{u} = 1 = -\underline{u};\ \overline{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -\underline{x}.$$

Example 2 (b)

$$\overline{u} = 1 = -\underline{u};\ \overline{x} = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} = -\underline{x}.$$

The tuning parameters are $Q = C^T C, R = 1, n_c = 2, n_G = 2$, Laguerre parameter $p = 0.38$, Kautz parameters $(a, b) = (0.38, 0.3)$ and $\gamma = \infty$.

Figure 5.3 and 5.4 show the region of attraction for KOMPC, LOMPC, ERPC and GERPC for the same number of d.o.f. i.e. $n_c = 2$. Figure 5.3 shows that KOMPC has a larger MCAS than LOMPC, GERPC and ERPC. Whereas in Figure 5.3, there are some initial points in the GERPC MCAS which are infeasible for LOMPC and KOMPC algorithms. Table 5.1 shows the MCAS volume comparison for $n_c = 2$, it is clear that KOMPC and LOMPC enlarge the region of attraction. The number of inequalities to describe the MCAS are shown in Table 5.2. The number of inequalities with parameterised algorithms (i.e. LOMPC and KOMPC) is insignificantly larger than for the ERPC and GERPC algorithm. Similarly to the previous Example 1(b), for the non-symmetric case, KOMPC and LOMPC enlarge the region of attraction significantly more than GERPC.

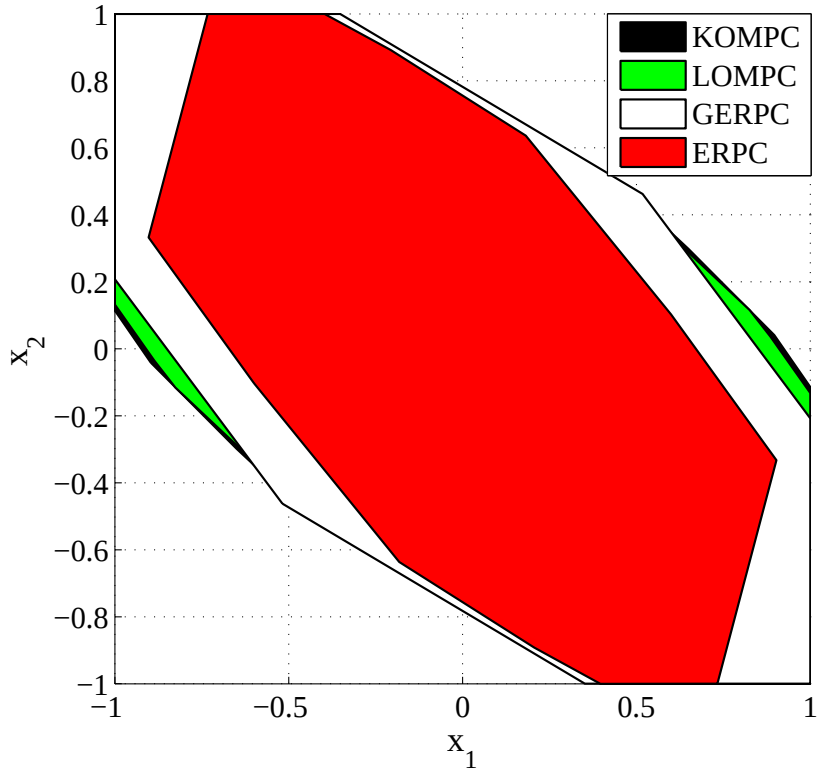

Figure 5.3: Comparison of MCAS for $n_c = n_G = 2$ for KOMPC, LOMPC, GERPC and ERPC algorithms for Example 2(a).

5.4.3 Example 3 - $x \in \mathbb{R}^3$

For this example the discrete-time state-space model and constraints are

$$A = \begin{bmatrix} 1.4 & -0.10504 & -0.1080 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}; \; B = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix}; \; C = \begin{bmatrix} 5 & 7.5 & 0.5 \end{bmatrix}; \tag{5.22}$$

Example 3 (a)

$$\overline{u} = 0.04 = -\underline{u}; \; \overline{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = -\underline{x}.$$

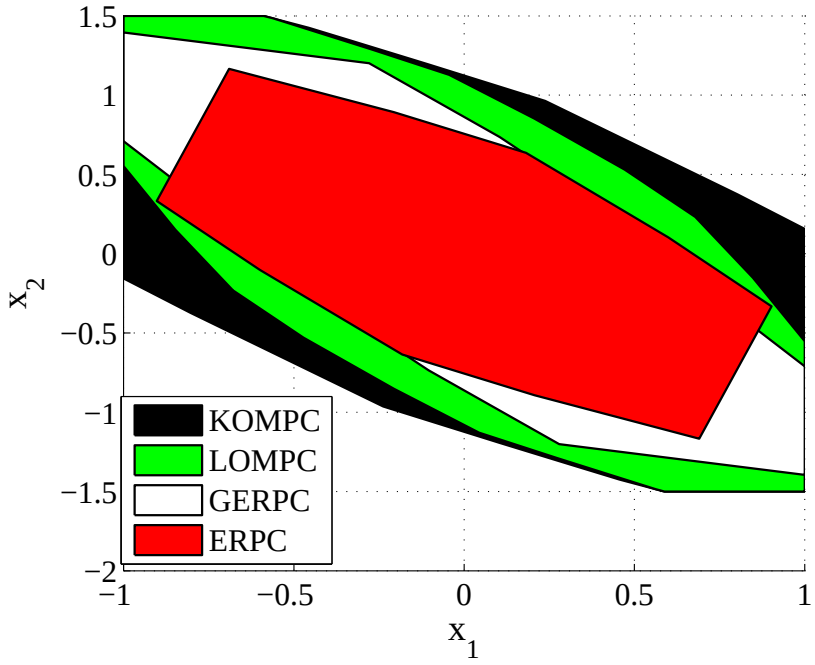

Figure 5.4: Comparison of MCAS for $n_c = n_G = 2$ for KOMPC, LOMPC, GERPC and ERPC algorithms for Example 2(b).

Table 5.1: MCAS volume comparison for $n_c = 2$

	KOMPC	LOMPC	GERPC	ERPC
Example 1(a)	3.8957	3.8813	3.8370	2.8388
Example 1(b)	4.6880	4.6859	2.7247	2.3444
Example 2(a)	2.6913	2.6812	2.6908	2.0914
Example 2(b)	4.1421	3.5964	2.6634	2.1473

Example 3 (b)

$$\overline{u} = 0.04;\ \underline{u} = -0.08;\ \overline{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = -\underline{x}.$$

The tuning parameters are $Q = C^T C, R = 1, n_c = 3, n_G = 3$, Laguerre parameter $p = 0.5$, Kautz parameters $(a, b) = (0.5, 0.49)$, GOMPC ($3rd$ order) $(a_1, a_2, a_3) = (0.5, 0.49, 0.48)$ and $\gamma = \infty$.

Table 5.3 shows the MCAS volume comparison for $n_c = 3$, it is clear that GOMPC using $3rd$ order dynamics enlarge the region of attraction for both symmetric and non-

Table 5.2: Comparison of number of inequalities to describe MCAS for $n_c = 2$

	KOMPC	LOMPC	GERPC	ERPC
Example 1(a)	83	117	124	18
Example 1(b)	105	120	170	30
Example 2(a)	29	34	22	23
Example 2(b)	39	46	44	28

symmetric cases. The number of inequalities to describe MCAS is shown in Table 5.4, for both cases, the number of inequalities for GOMPC, KOMPC and LOMPC algorithms are insignificantly larger than ERPC algorithm.

5.4.4 Example 4 - $x \in \mathbb{R}^4$

For this example the discrete-time state-space model and constraints are

$$A = \begin{bmatrix} 0.900 & -0.105 & -0.108 & 0.200 \\ 0.600 & 0 & 0 & -0.100 \\ 0 & 0.800 & 0 & 0.300 \\ 0 & 0 & 0.800 & 0 \end{bmatrix};\ B = \begin{bmatrix} 0.2 \\ 0 \\ 0 \\ 0.5 \end{bmatrix};\ C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}; \tag{5.23}$$

Example 4 (a)

$$\overline{u} = 1 = -\underline{u};\ \overline{x} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} = -\underline{x}.$$

Example 4 (b)

$$\overline{u} = 1;\ \underline{u} = -2;\ \overline{x} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} = -\underline{x}.$$

The tuning parameters are $Q = C^T C, R = 1, n_c = 4, n_G = 4$, Laguerre parameter $p = 0.7458$, Kautz parameters $(a, b) = (0.3251, 0.7458)$, GOMPC ($3rd$ order) $(a_1, a_2, a_3) = (0.321, 0.7458, 0.6595)$, $(a_1, a_2, a_3, a_4) = (0.321, 0.7458, 0.6595, 0.7755)$ and $\gamma = \infty$.

Table 5.3: MCAS volume comparison

	GOMPC (4th order)	GOMPC (3rd order)	KOMPC	LOMPC	GERPC	ERPC	n_c
Example 3(a)		3.9998	2.3469	1.8449	0.8877	0.1366	3
Example 3(b)		4.0	3.1812	2.7025	0.8880	0.3072	3
Example 4(a)	158030	157830	157670	157660	144150	99931	4
Example 4(b)	158030	157930	157930	157930	152840	128700	4

Table 5.4: Comparison of number of inequalities to describe MCAS

	GOMPC (4th order)	GOMPC (3rd order)	KOMPC	LOMPC	GERPC	ERPC	n_c
Example 3(a)	-	32	24	26	29	24	3
Example 3(b)	-	28	22	26	22	26	3
Example 4(a)	49	52	54	66	41	34	4
Example 4(b)	50	56	57	63	36	35	4

Table 5.3 shows the MCAS volume comparison for $n_c = 4$, it is clear that GOMPC using $4th$ order dynamics enlarges the region of attraction for both the symmetric and non-symmetric cases. However, calculating the polyhedra using the GOMPC algorithm reduces the number of constraints compared to both KOMPC and LOMPC. The number of inequalities to represent the MCAS of GOMPC is larger as compared to GERPC and ERPC.

Regions of attraction

Regions of attraction are compared in Figure 5.1, 5.2, 5.3 and 5.4, and MCAS volume is compared in Table 5.1 and 5.3 using the Multi-Parametric Toolbox (MPT) [123]. Alternative parameterisations (i.e. LOMPC, KOMPC & GOMPC) noticeably enlarged the region of attraction compared to GERPC and ERPC. For non-symmetric constraints, alternative parameterisation enlarges region of attraction significantly even compared to GERPC with $\gamma = \infty$.

The LOMPC is tuned using the eigenvalues of the closed loop stable system, that is the one that arises from the feedback gain K in (4.2). The Kautz dynamics are selected to be equal or in the vicinity of LOMPC pole, $3rd$ order dynamics are selected to be equal are in vicinity of Kautz dynamics and similarly $4th$ order using $3rd$ order dynamics. Therefore it is observed from Figure 5.1, 5.2 and 5.3 that there is not much difference in the

Table 5.5: Comparison of performance index

Example	Initial states	GOMPC (4th order)	GOMPC (3rd order)	KOMPC	LOMPC	ERPC
1(a)	(-0.4,0.5)	-	-	18.20	18.20	18.20
1(b)	(-0.4,0.1)	-	-	16.60	16.60	16.60
2(a)	(0.7,-0.9)		-	18.26	18.26	17.38
2(b)	(-0.2,-0.6)		-	35.93	36.07	35.04
3(a)	(-0.1,0.2,-0.9)	-	126.55	126.10	125.97	125.84
3(b)	(-0.1,-0.7,0.9)	-	256.68	256.15	256.15	256.13
4(a)	(9,5.7,-9,-9)	26.51	26.51	26.51	26.51	26.51
4(b)	(9,7.9,-8.2,9)	47.44	47.44	47.44	47.44	47.44

resulting region of attractions between LOMPC and KOMPC. Systematic mechanisms will be discuss in Chapter 6 to choose the best tuned alternative dynamics to further enlarge the region of attraction.

Closed loop performance comparison

Closed loop performance comparison is based on computing performance $J_k = \sum_{i}^{\infty}[x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R x_{k+i|k}]$ for all algorithms, where $\{x_{k+i|k}, i \geq 0\}$ is a predicted trajectory of $x_{k+1} = Ax_k + Bu_k$ with $x_{k|k} = x_k$. The performance of the algorithms is contrasted for different initial states. To make the comparison more meaningful therefore it is advisable to consider the initial conditions which lie inside the regions of attraction of the all the algorithms under consideration. In Table 5.5, for example 1, performance is calculated near the boundary of the region of attraction of the ERPC algorithm. It is interesting to observe that GOMPC, KOMPC, LOMPC and LOMPC enlarge the region of attraction without too much degrading the closed loop performance.

Computational load comparison

The computational load comparison is based on the number of inequalities with similar d.o.fs.. Table 5.2 and 5.4 shows the number of equalities using $n_c = n_x$ d.o.f. for GOMPC, KOMPC, LOMPC, GERPC and ERPC algorithms. The alternative algorithms enlarge the region of attractions at the price of an increase in the number of inequalities. In many cases, the number of inequalities for the alternative algorithms are slightly larger than the ERPC algorithm. The number of inequalities may be reduced without compromising

the region of attraction using higher order dynamics however it is recommended that $n_G \leq n_c$.

Summary

The GOMPC provides a higher dynamic parameterisation than LOMPC and KOMPC which can be use to improve the trade off between the region of attraction, performance and computational load. There are few pros and cons of GOMPC.

- GOMPC provides a generalised function parameterisation within existing predictive control algorithms.
- It tackles the question related to possible alternative choices, what are they and how are they introduced into existing algorithms?
- It improves the region of attraction as compared with ERPC, GERPC, LOMPC and KOMPC without increasing n_c.
- GOMPC provides an alternative proposal to the existing systematic approach developed in [25].
- GOMPC provides designers with more tuning choices as compared with KOMPC and LOMPC. The next chapter will propose a systematic selection of generalised function parameterisation.
- The computational burden introduces a hard limit on the selection of order of higher order function parameterisation i.e. $n_G \leq n_c$. The generalised function parameterisation enlarges the region of attraction without increasing the number of d.o.f. (or n_c).
- The higher order function parameterisation also introduces more parameter variable(s) to tune, to simplify the trade off between region of attraction and performance.

5.5 Conclusions

This chapter has argued for the potential benefits of generalised functions as an alternative parameterisation for maximising the region of attraction in conventional MPC algorithms. Laguerre and Kautz functions are presented as a special case of generalised

functions. The mathematical representation of the generalised network is presented using a state space model with an example. It has been shown through numerical examples that in many cases the region of attraction can be improved without too much degrading performance. However, of more significance, the chapter has tackled the question concerning the earlier proposed use of Laguerre functions to parameterise the d.o.f. in the prediction and in fact, this chapter indicates that in many cases the higher order functions may indeed be preferable in general. This benefit cannot be proven generally and in some cases it is not required. The next chapter will tackle the question of whether there exists a 'systematic' method for choosing the best parameterisation dynamics for any given problem.

Chapter 6

A systematic selection of an alternative parameterisation

This chapter presents an **original contribution** to the thesis. The focus of this chapter to make contributions in the area of predictive control and in particular examines to what extent different methods for parameterising the degrees of freedom within the input trajectories can improve aspects of the region of attraction and performance. A few earlier papers [2, 18, 26] have suggested the potential for the Laguerre functions to be effective within a predictive control design, but without giving explicit design guidelines. This chapter extends that work by looking not only at systematic choices for the Laguerre parameters but also other choices of orthonormal functions. Systematic mechanisms are discussed to choose the best tuned alternative parameterisation dynamics. The efficacy of the proposed parameterisations within existing predictive control algorithms are demonstrated by examples.

This chapter is organised as follows: Section 6.1 presents the introduction and motivation of the chapter; Section 6.2 proposes schemes to identify the best tuned parameterised dynamics based on multi-objective optimisation and on pragmatic selection rules. Numerical simulations and the results are in Section 6.3 and the chapter finishes with conclusions in Section 6.4.

6.1 Introduction

All MPC algorithms to some extent allow a trade off between performance, the regions of attraction and the complexity of the optimisation, and require further study to handle this trade off. One important issue for real time implementation on low cost processors (with limited processing capabilities) is the computational burden [203]; it is simply not possible to code and implement an extensive optimisation algorithm and thus simpler optimisers need to be used. In recent years research in this area has focused on parametric solutions [118] or fast optimisation [127]. However, this chapter follows a different route and instead considers the underlying structure of MPC algorithm. Conceptually the main question is whether changing the structure may simplify the associated optimisation e.g. [171]. Specifically, this chapter proposes a systematic selections of parameterisations for the input predictions with the specific goal of enabling a large region of attraction without requiring large numbers of degrees of freedom, and obviously without a significant detriment to closed loop performance.

The focus of this chapter is on the choice of the d.o.f. in a conventional OMPC type of algorithm. A set of sequences known to have been effective in earlier studies [18, 26] are the Laguerre functions and thus a logical route of further study was to investigate this class of functions in more detail. Previous chapters demonstrated the potential of higher order orthonormal functions and thus this chapter will bring all the findings together and also propose systematic guidelines for choosing appropriate orthonormal functions for use in MPC design.

The key question left to be resolved is, what are the possible choices and how to select a best alternative parameterisation to allow for a large region of attraction without detriment to the performance and computational burden? This chapter focusses on Laguerre, Kautz and higher order orthonormal functions. A generalised selection algorithm based on this suite of parameterisations is proposed and thus extends earlier studies. Some discussion is also given to the difference between 'optimal selection' methods and pragmatic choices which can be made without an associated offline computation.

6.2 The best parameterisation selection

Having proposed that a generalised parameterisation offers a systematic tool for creating a flexible prediction structure that works well within optimal MPC, the remaining question is how does one best deploy this flexibility to achieve the desired trade off between MCAS volume, complexity and performance? Worded another way, what is the optimal choice

of parameterisation dynamics H_G, which of course also implies the need to define what we mean by optimal choice. From equation (5.5), in generalised parameterise dynamics there are two main choices within the future input predictions.

1. Select the order of prediction dynamics that is the number of poles a_i in A_G.
2. Select specific values for the poles a_i.

This chapter, in the following sections, proposes two systematic approaches for selecting and tuning the parameterisation dynamics.

6.2.1 Dynamic order selection

The main focus of parameterised optimal MPC is to improve the region of attraction with limited number of d.o.f. or with inexpensive optimisation. Higher order parameterised dynamics have more flexibility in choosing dynamic parameters to overcome the trade off between the region of attraction and closed loop performance loss. However, there is a commonsense observation that $n_c \geq m$, that is to fully utilise the flexibility in having m poles of generalised function dynamics, one should use at least m d.o.f.. Consequently, where one knows that a given value of n_c is sufficient, it is not recommended to use a higher number of poles. Having said that, the normal value of n_c in OMPC required to get close to global region of attraction is relatively high (in this thesis approximated using $n_c = 20$) and thus in practice the expected choices of $dim(A_G)$ are much smaller and an objective choice can only be made with some offline analysis - as discussed in the next subsection.

6.2.2 Selection of parameterisation poles using a multi-objective approach

The proposal here is to use analysis tools from the multi-objective optimisation community [204]; this is because there are multiple objectives of performance, MCAS volumes and computational burden. Each of these objectives needs to be quantified in terms of the parameterisation parameters $\alpha = [a_1, \ldots, a_m]$ (where $0 \leq a_i < 1$) and then the optimisation is formulated in terms of trade off between the region of attraction and performance with a limited number of d.o.f.. For simplicity, this chapter considers the trade off with a fixed value of n_c and thus asks what is the best one can get for a given

order of optimisation. Clearly increasing or decreasing n_c will have an obvious impact and one could judge, for a given n_c, whether the best available is satisfactory.

This multi-objective optimisation is defined in terms of maximising the MCAS volume ν and closed loop performance β using a monte-carlo sampling approach. Consequently computations are based on choosing a number of points $x = (x_1, \ldots, x_n)$ equi-spaced (by solid angle for 2-dimensional systems) or random selection or chosen uniformly on the unit hyper-sphere and then scaling to be on the appropriate outer boundary of the MCAS.

1. Define $\mathcal{P}_{opt} = \{(x, c)|Mx_k + N \underset{\rightarrow}{c}_k \leq d\}$ as the Global MCAS (OMPC for a large n_c, notionally this is taken to be about $n_c = 20$ for the examples in this chapter but in principle could be any appropriate value). This is taken to represent the global 'largest' possible region of attraction.

2. Define the MCAS for the proposed parameterisation using $\mathcal{P}_H = \{(x, \rho)|Mx_k + NH_G \underset{\rightarrow}{\rho}_k \leq d\}$, that is the polytope sliced by the parameterised matrix H_G. [Note for clarity there has been some abuse of the terms M, N here as they can be the same for $\mathcal{P}_{opt}, \mathcal{P}_H$ only if in non-minimal form.]

3. $vol(\pi\mathcal{P}_{opt})$ and $vol(\pi\mathcal{P}_H)$ represent the MCAS volumes, where $\pi P = \{x|\exists u\ (x, u) \in P\}$ is the projection operation. As these can be somewhat time consuming to compute for high dimensional polyhedrals, they are approximated by the average of the distance from the origin to the boundary of the associated MCAS is determined by solving a linear programming (LP), for each chosen point x_i; clearly the larger the distance better the feasibility.

4. The predicted performance, for given points x_i are represented by the optimised values of $J_{opt,k}(x_i)$, $J_{H,k}(x_i)$ from equations (4.20,5.12) for Global OMPC and GOMPC respectively. To ensure fairness, the comparison uses the scaling of a given direction x_i which is feasible for all methods being compared.

5. Define ν and β as:

$$\nu = \frac{vol(\pi\mathcal{P}_H)}{vol(\pi\mathcal{P}_{opt})},\ \beta = \frac{1}{n}\sum_{i=1}^{n}\frac{J_{opt,k}(x_i)}{J_{H,k}(x_i)}, \tag{6.1}$$

The multi-objective optimisation can now be summarised as:

$$\begin{aligned} J_k(n_c, n) := \min_{\alpha} \beta, \max_{\alpha} \nu \quad s.t. \quad & Mx_k + NH_G \underset{\rightarrow}{\rho}_k \leq d, \\ & \alpha = [a_1, \ldots, a_m], \quad 0 < a_i < 1, \quad m \leq n_c, \end{aligned} \tag{6.2}$$

This optimisation results in a pareto surface between the d.o.f. n_c, the resulting size of region of attraction ν and the average performance β. These curves may be used to identify the best parameterisation selection with tuned parameter(s).

Remark 6.2.1. Average performance comparison β has limited value because one can only contrast within the MCAS of all algorithms to be compared, as once beyond its region of attraction, an algorithm is undefined. The OMPC value is used to assess $J_{H,k}(x_i)$ for fair comparison.

There are some limitations using multi-objective optimisation:

- A systematic design rule is proposed to choose the best tuned generalised function parameterisation dynamics using multi-objective optimisation. The multi-objective optimisation has inherent issues like heuristic algorithms, non-convexity, convergence etc.
- Solution uniqueness is an open question regarding multi-objective optimisation, as it is typical for the result of the optimisation to be a pareto-optimal family of solutions.
- The multi-objective optimisation requires more offline computational effort to simplify the trade off between performance, region of attraction and computational load.
- An approximation of Global MPC (OMPC with for a large n_c) is used to represent the global 'largest' possible region of attraction. It provides a bound on how much enlargement of the region of attraction can be obtained for a given system.
- The closed-performance comparison is contrasted for initial conditions which lie inside the regions of attraction of all algorithms, since the region of attraction differ in size for different algorithms.
- The computation of ν is approximated using the mean distance not the volume of projections i.e. $vol(\pi P)$.

6.2.3 Pragmatic Selection

It is recognised that multi-objective optimisations can been intractable, albeit offline. So, although these offer good insight into the trade offs and thus what can be achieved, it may not be a useful tool for the average engineer who wants more simplistic but effective design

guidance. This section proposes some pragmatic selections to identify the parameterised dynamics which, albeit with sub optimal parameter value(s), are likely to be close enough to the best choices to be almost equivalent in a real uncertain scenario.

These selections are based on the underlying closed loop stable system, that is the one that arises from the feedback K in (4.2). The pole location(s) of parameterised dynamics can be selected to be equal to or in vicinity of pole(s) of the optimal closed loop system; this may provide a good starting point. The author makes no claim that this can be proven in any objective sense, but it is based on observation results from numerous tests using the multi-objective approach.

Remark 6.2.2. Pragmatic selection is based on closed loop stable system poles, so this selection assumes that the parameterisation dynamics are of lower order or equal to the system dynamics. If lower order, one would focus on the dominant poles.

6.3 Numerical examples

In this section numerical examples are presented to illustrate how the proposed approaches to the parameterisation of the input predictions within OMPC perform. The aim is to compare the trade off between average MCAS volume gain, average performance and the number of d.o.f. (or computational burden). The OMPC algorithm with an equivalent number of d.o.f. is used as a benchmark for comparison, along with the Global OMPC (global optimum achieved using OMPC with $n_c = 20$). For convenience LOMPC assumes one parameterised pole, KOMPC two parameterised poles and GOMPC three parameterised poles; although higher numbers are possible with more dynamics involved.

The efficacy of the pragmatic selection for LOMPC, KOMPC and GOMPC algorithm in comparison with ERPC [21] and GERPC [25] is presented using randomly selected numerical examples for the linear time invariant case. The main focus is to present a statistical analysis based on the enlargement of the region of attraction. The comparison is based on the highly tuned GERPC algorithm with pragmatic choices of LOMPC, KOMPC and GOMPC. The pole locations of parameterised dynamics are selected to be equal to or in the vicinity of pole(s) of the of the closed loop stable system, that is the one that arises from the terminal control law.

6.3.1 Optimal selection based on multi-objective optimisation solution

The goal is to produce trade off curves between the resulting average MCAS volume gain ν and the average performance β as a function of the different parameterisation parameter(s) and d.o.f. or n_c. In simple terms the best strategy has a high value of ν and β. The multi-objective procedure was run for different initial conditions $x = [x_1, \ldots, x_n]$ (i.e. for $2nd$ order example $n = 36$, $3rd$ order example $n = 100$ and $4th$ order example $n = 200$), which were chosen uniformly on the unit hyper-sphere. The resulting trade off plots and the parallel coordinates are shown to represent different variation of parameter(s) α and effect on both β and ν.

Parallel coordinate plots in Figure 6.1, Figure 6.6, and Figure 6.10 are a common way for visualising a set of points in higher dimensional space consisting of parallel lines (i.e. a_1, a_2, a_3, ν, and β) using multi-objective optimisation analysis, typically vertical and equally spaced. The vertical parallel axis containing α varies from $0 \leq a_i < 1$, the average MCAS volume gain ν varies from $0 \leq \nu \leq 1$ and average performance β varies from $\beta > 0$. The parallel coordinates are used to identify the best tuned parameter(s) dynamics with $\beta \geq 1$ and $\nu \approxeq 1$. Simulation results with fewer parameter(s) selection lines are shown in the figure for GOMPC ($3rd$ order), KOMPC and LOMPC only for $n_c = 3$, but there are similar results for different variation of n_c. The parallel plots in Figure 6.1, 6.6, and 6.10 shows the results of the optimisation in terms of parameter variations to be a pareto-optimal family of solutions. The multi-objective optimisation is done using NSG-II Matlab toolbox.

Figure 6.2, Figure 6.7, and Figure 6.11 show the pareto points between ν and β for GOMPC ($3rd$ order), KOMPC and LOMPC as a function of different parameter(s) with $n_c = 3$. The trade off between ν and β represents the effect of different parameter(s) selection on both performance and MCAS volume, the optimal solution is selected from the pareto-optimal family solutions with maximum MCAS volume gain (i.e. $\nu \approxeq 1$) and minimum performance drop (i.e. $\beta \geq 1$). The colour circle on the trade off plot represents the pragmatic selection based on closed loop stable eigenvalues. The resulting optimal selection for GOMPC ($3rd$ order), KOMPC and LOMPC are compared for both ν and β in Figure 6.3, Figure 6.4, Figure 6.8, Figure 6.9, Figure 6.12 and Figure 6.13 as a function of $n_c = [2, \ldots, 8]$.

6.3.2 Pragmatic selection

The pragmatic selection of parameterisation dynamics uses the closed loop eigenvalues. The selection of parameterisation dynamics dimension has an upper limit based on the

closed loop eigenvalues. A colour circle shown on the trade off plots represent the pragmatic selections. It is clear from Table 6.1, Figure 6.2, Figure 6.7, and Figure 6.2 that in many cases pragmatic choice is a suboptimal solution, it is in vicinity of an optimal pareto solution.

6.3.3 Example 1

$$A = \begin{bmatrix} 0.9 & -0.1 \\ 0.05 & 1 \end{bmatrix};\ B = \begin{bmatrix} 0.3 \\ 0.1 \end{bmatrix};\ C = \begin{bmatrix} 0 & 1 \end{bmatrix};$$
$$-0.5 \leq u_k \leq 1;\ |\Delta u_k| \leq 0.4;\ -\begin{bmatrix} 5 \\ 10 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix};\ Q = diag(1,1);\ R = 0.1.$$

Figure 6.1 shows the parallel coordinates between different parameter(s) selection, β and ν using $n_c = 3$ for GOMPC ($3rd$ order), KOMPC and LOMPC. Figure 6.2 shows the parameter variations for all alternative algorithms achieve global MCAS volume with negligible performance loss on average. The trade off plot also shows the pragmatic choice with 96% of the global MCAS with negligible performance loss (less than 1% worse on average). This trade off and parallel coordinates plot may be used as parameter(s) selection criteria. Figure 6.5 shows the pareto-optimal family of solutions for Laguerre and Kuatz parameterisation dynamics using $n_c = 2$ to simplify the trade off between the region of attraction and improved performance.

The optimal parameter(s) were selected with global MCAS volume using parallel coordinates and pareto points (with $\nu \cong 1$ and $\beta \geq 1$) in Figure 6.2. The LOMPC (with $p = 0.8080$), KOMPC (with $a = 0.8740, b = 0.0373$) and GOMPC (with $a_1 = 0.9224, a_2 = 0.0289, a_3 = 0.5430$) was run with optimal tuning parameters by varying $n_c = (1, \ldots, 8)$ and results are shown in Figure 6.3 and Figure 6.4 for a set of 36 feasible initial conditions. Alternative algorithms (i.e. LOMPC, KOMPC and GOMPC) achieved global MCAS volume with 3 d.o.f. with maximum 0.29% performance loss on average. For $n_c = 2$, GOMPC reach, to within less than 4%, both KOMPC and LOMPC reach, to within less than 8%, the MCAS for OMPC with $n_c = 20$. KOMPC and GOMPC enlarge the region of attraction with negligible performance loss and less number of inequalities as compared to LOMPC.

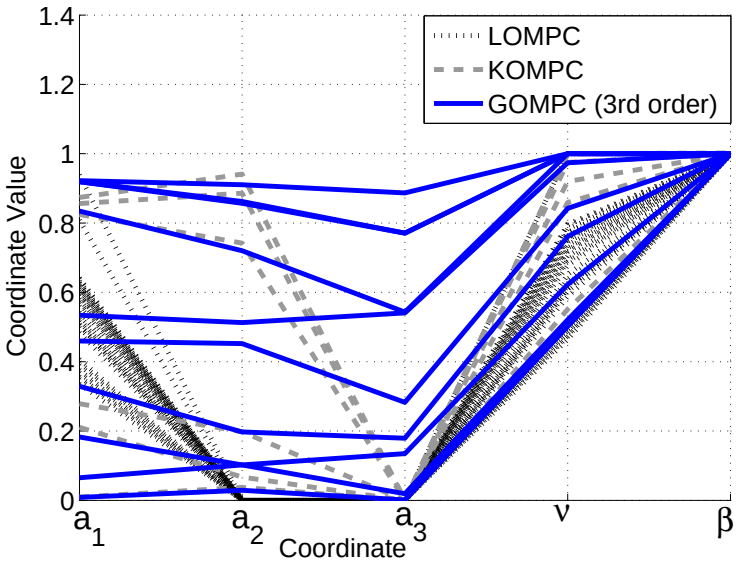

Figure 6.1: Comparison of parallel coordinates for Example 1.

6.3.4 Example 2

$$A = \begin{bmatrix} 0.9146 & 0 & 0.0405 \\ 0.1665 & 0.1353 & 0.0058 \\ 0 & 0 & 0.1353 \end{bmatrix};\ B = \begin{bmatrix} 0.054 & -0.075 \\ 0.05 & 0.147 \\ 0.8647 & 0 \end{bmatrix};\ C = \begin{bmatrix} 1.799 & 13.216 & 0 \\ 0.823 & 0.5 & 0.1 \end{bmatrix};$$

$$|u_k| \leq 1;\ |\Delta u_k| \leq 0.5;\ |x_{i,k}| \leq 1;\ Q = diag(1,0,0);\ R = I.$$

This example seems to be quite simple to achieve both global MCAS and near optimal solutions with low degree of freedom. Figure 6.8 and Figure 6.9 show the average MCAS gain and average performance by varying d.o.f. for a set of 100 feasible initial conditions. The resulting parallel coordinates and trade off curves are shown in Figure 6.6 and Figure 6.7. The parameter(s) variation for KOMPC and GOMPC achieve 100% of global MCAS with approximately global optimal performance on average. LOMPC achieves approximately global MPC with less than 1% worst performance on average. Whereas the pragmatic selection also achieves 100% of global MCAS with 1% worst performance on average.

The LOMPC (with $p = 0.6306$), KOMPC (with $a = 0.6651, b = 0.0438$) and GOMPC (with $a_1 = 0.7616, a_2 = 0.6317, a_3 = 0.0938$) was run with optimal tuning parameters by varying $n_c = (2, \ldots, 8)$ and results are shown in Figure 6.8 and Figure 6.9. It is observed

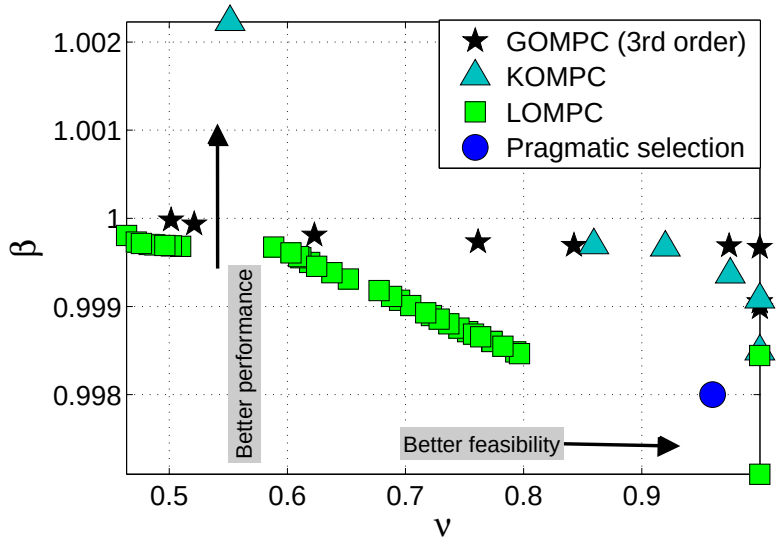

Figure 6.2: Comparison of pareto front for Example 1.

from Figure 6.8 and Figure 6.9 that optimal tuning may achieved low dimensional parameterisation dynamics with improved trade off. GOMPC (3*rd* order) and KOMPC gets within global region of attraction with just 3 d.o.f.. GOMPC (3*rd* order) and KOMPC are good choices with global MCAS and approximately optimal performance with less number of inequalities as compared with LOMPC.

6.3.5 Example 3

$$A = \begin{bmatrix} -1.80 & 0 & 0 & 0.9 \\ 0.386 & -0.406 & 0 & 0.9 \\ 0 & 0 & -0.60 & 0 \\ 0.81 & -0.770 & 0.45 & -0.406 \end{bmatrix};\ B = \begin{bmatrix} -0.4326 & 0 \\ 0 & 1.1909 \\ 0.1253 & 1.1892 \\ 0.2877 & -0.0376 \end{bmatrix};$$

$$C = \begin{bmatrix} -0.9080 & 0.6967 & 0.2899 & -0.1285 \\ 0.8901 & -0.5394 & 0 & -0.7538 \end{bmatrix}; |u_{i,k}| \leq 1;\ |\Delta u_{i,k}| \leq 0.5;\ |x_{i,k}| \leq 5;$$

$$Q = diag(1,1,1,1);\ R = I.$$

The parameter(s) variations using trade off curves and parallel coordinates are shown in Figure 6.10 and Figure 6.11 achieves 96% of global MCAS volume with improved performance on average. Whereas the pragmatic selection achieves 95% of the global

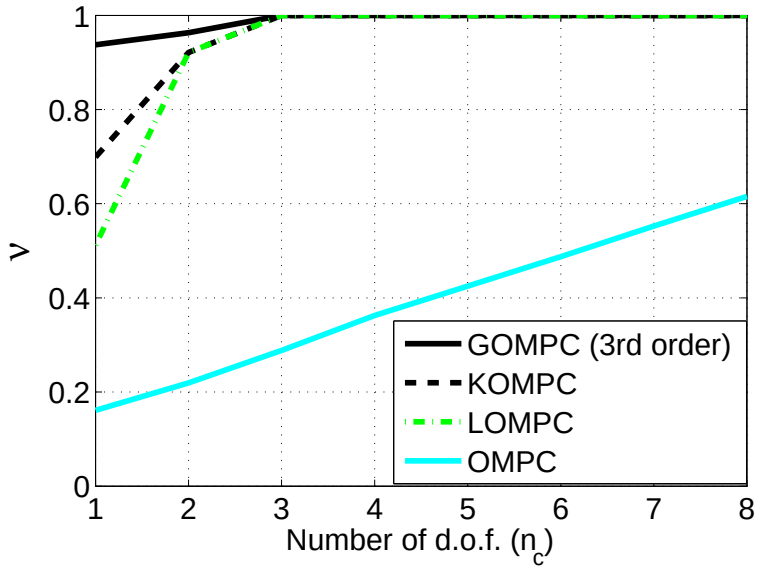

Figure 6.3: Comparison of average MCAS volume for Example 1.

MCAS with $\beta = 0.98$. It is observed from Figure 6.11 that for parameterised algorithms the pareto-optimal family of solutions are similar and Laguerre parameterisation dynamics are enough to simplify the trade off between the region of attraction and improved performance.

The LOMPC (with $p = 0.1808$), KOMPC (with $a = 0.013, b = 0.4188$) and GOMPC (with $a_1 = 0.013, a_2 = 0.0294, a_3 = 0.4428$) was run with optimal tuning parameters by varying $n_c = (2, \ldots, 8)$ and results are shown in Figure 6.12 and Figure 6.13 for a set of 200 feasible initial points. It is interesting to observe from Table 6.1 that the parameterised algorithms achieved little more as compared to OMPC using optimal tuning. Similarly as in the previous examples, GOMPC ($3rd$ order) is a good choice with improved performance and a larger MCAS compared to LOMPC and KOMPC.

6.3.6 Computational load and performance comparisons

The computational load is assumed here to depend upon the number d.o.f. and the number of inequalities[1]. It is assumed that the MCAS representations are reduced to minimal form which in itself is a major computational load; without such a step one would expect the number to be similar in all cases. The number of inequalities is compared for all algorithms with $n_c = 3$ and are summarised in Table 6.1.

[1]Investigations into the impact of structure on the computational loading are ongoing

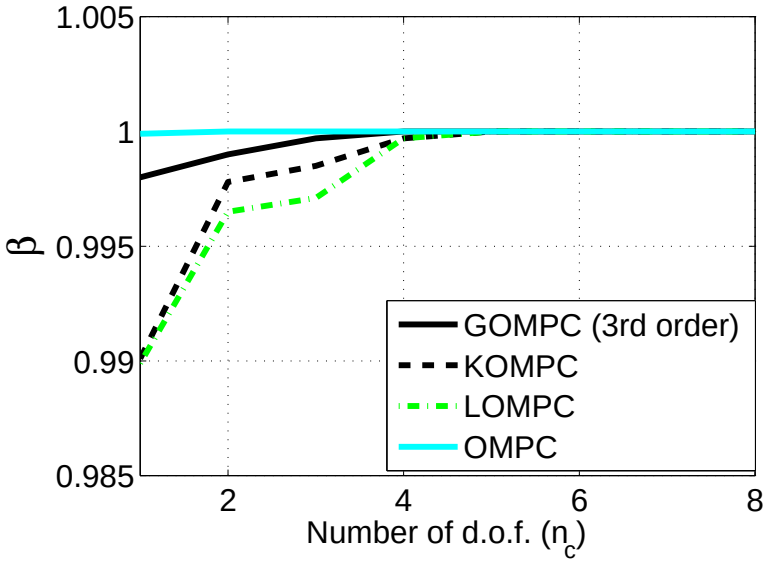

Figure 6.4: Comparison of average performance for Example 1.

The alternative parameterised algorithms (that is with different numbers of poles a_i) have similar numbers of inequalities, but notably these are more than for OMPC. However, KOMPC and GOMPC have a nearly same number of inequalities.

Performance comparisons will be based on computing average performance β over the closed loop responses. The lower the performance drop i.e. $\beta \geq 1$, the better the performance. However, comparisons are meaningful only if the initial conditions are within the region of attraction of all algorithms to be compared as once beyond its region of attraction, an algorithm is undefined. Therefore average normalised closed loop performance for all algorithms with $n_c = 3$ are summarised in Table 6.1; normalisation is done against

Table 6.1: Number of inequalities and normalised average runtime cost using $n_c = 3$ required by OMPC, LOMPC, KOMPC and GOMPC applied to all examples models.

Algorithm	Example 1			Example 2			Example 3		
	Ineq.	β	ν	Ineq.	β	ν	Ineq.	β	ν
OMPC	90	1	0.3	173	0.9865	0.8541	122	0.9717	0.9500
LOMPC	172	0.9971	1	309	0.9910	0.9996	184	0.9845	0.9586
KOMPC	160	0.9985	1	305	0.9979	1	203	0.9881	0.9625
GOMPC	160	0.9997	1	310	0.9981	1	206	0.9902	0.9645
Pragmatic Selection	157	0.9980	0.96	320	0.9906	1	217	0.9885	0.9500

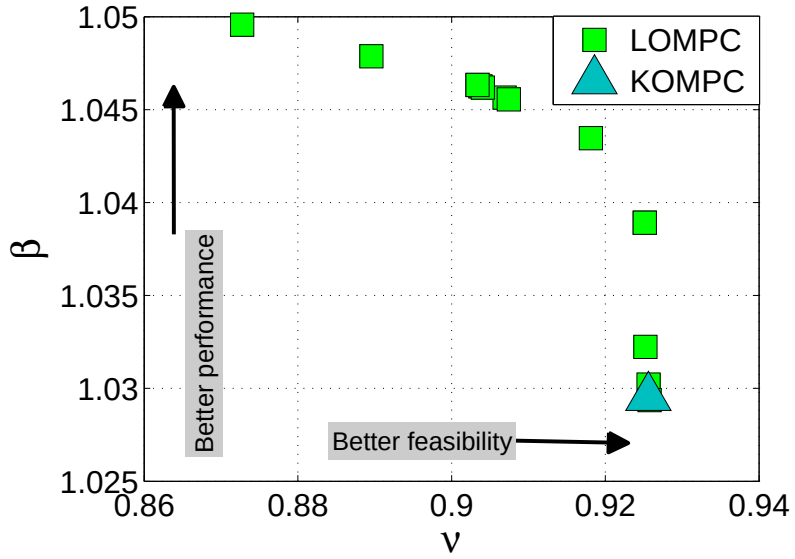

Figure 6.5: Comparison of pareto front for Example 1 using $n_c = 2$.

Global OMPC. In Example 1 and 2 alternative parameterisations have achieved global MCAS with less than 1% performance drop on average, whereas in Example 3 there is little more achieved as compared to OMPC. In Example 3, GOMPC achieved more than 96% of global MCAS with less than 1% worst performance on average.

6.3.7 Summary

The numerical examples show that in many cases alternative parameterisations enlarge the region of attraction with little detriment to the closed loop performance.

- Systematic selection mechanisms (using multi-objective optimisation and pragmatic selection) of the parameterisation provide a tool to overcome a trade off between performance, region of attraction and inexpensive optimisation.
- For higher order generalised function parameterisations have more flexibility in parameter selection to overcome the trade off between the region of attraction and closed loop performance. However there is a limit on the selection of a higher order as from (5.5), $n_c \propto n_G$. If $n_G > n_c$, then this selection will compromise the computational burden.
- In many cases, pragmatic selection provides a good way to select the order of generalised function with suboptimal parameter(s) choice.

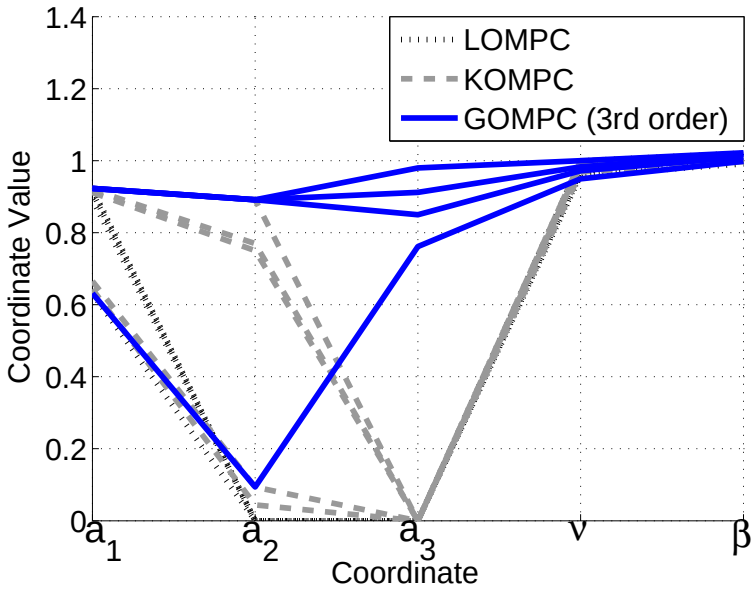

Figure 6.6: Comparison of parallel coordinates for Example 2.

An interesting observation from the trade off curves and parallel coordinates plots in Figure 6.2, 6.1, 6.7, 6.6, 6.11 and 6.10 is that systematic parameterisation selections often improve the trade off between the region of attraction, performance and computational burden. Another interesting observation is about the pragmatic selection, it is not an optimum choice but in many cases provides a good starting point to tune the parameterised dynamic.

6.3.8 Statistical analysis of pragmatic selection

The pragmatic selection is analysed further in this section using a statistical analysis for alternative parameterisation algorithms (i.e. LOMPC, KOMPC and GOMPC). The prime interest is to compare the enlargement of the region of attraction using LOMPC, KOMPC, GOMPC, GERPC [25] and ERPC [21] algorithms. The comparisons are based on 500 random systems with $x \in \mathbb{R}^2$, $x \in \mathbb{R}^3$, and $x \in \mathbb{R}^4$ (total of 1500 systems) using $n_c = n_x$.

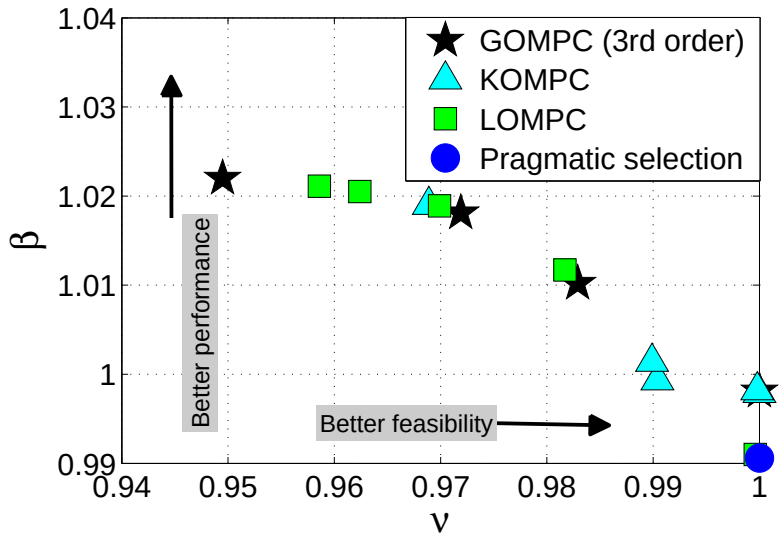

Figure 6.7: Comparison of pareto front for Example 2.

Example 4 – $x \in \mathbb{R}^2$

Consider 500 single input single output $2nd$ order (i.e. $x \in \mathbb{R}^2$) random systems subject to input and state constraints

$$-1 \leq u_k \leq 1; \quad -\begin{bmatrix} 4 \\ 1.5 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 4 \\ 1.5 \end{bmatrix}. \tag{6.3}$$

The tuning parameters are $Q = C^T C$, $R = 1$, $n_c = 2$ and $\gamma = \infty$. For this example, Kautz and Laguerre dynamics are used as alternative parameterisation dynamics (because $n_G \leq n_c$) and tuned using the pragmatic selection. The region of attractions are calculated using the Multi-Parametric Toolbox (MPT) [123].

Figure 6.14 shows the histograms of the MCAS volume for KOMPC, LOMPC, GERPC and ERPC. The histogram suggests that in many cases KOMPC and LOMPC have a larger MCAS volume than GERPC. Table 6.2 shows some statistics regarding the simulations. As shown, KOMPC has a larger MCAS volume on average than LOMPC, GERPC and ERPC.

This is further analysed using a box and whisker diagram in Figure 6.15 for MCAS volume comparison. In box and whisker diagram, the length of the box represents the inter quartile range (IQR), this is difference between the $25th$ and $75th$ percentiles of MCAS volume and horizontal lines in the box represent the median MCAS volume for

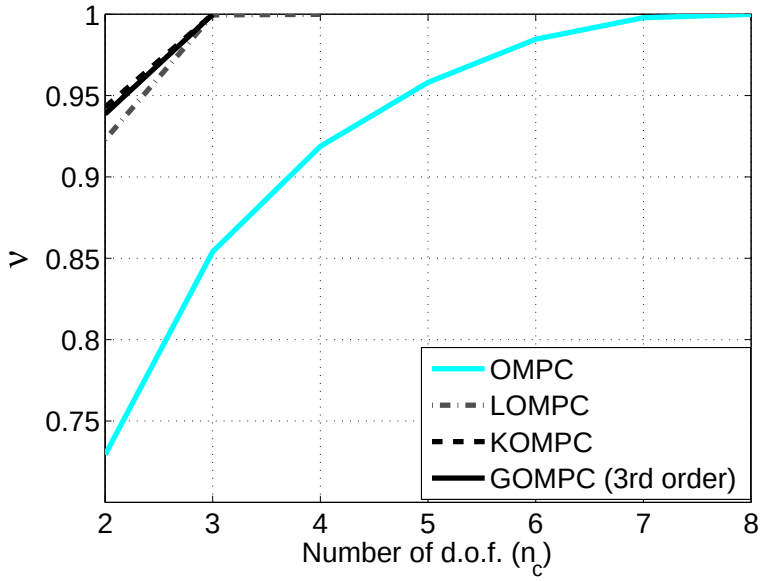

Figure 6.8: Comparison of average MCAS volume for Example 2.

random systems. The line outside the box represent the maximum and minimum values of MCAS volume (i.e. 1.5 times IQR range from the top and bottom of the box) and outliers are shown using '+' symbol. The box and whisker plot suggests that KOMPC and LOMPC enlarge the region of attraction further than GERPC. The outlier suggests that in some cases GERPC may be better than both LOMPC and KOMPC.

Table 6.2: Statistical results for MCAS volumes.

Algorithm	Example 4		Example 5		Example 6	
	Mean	Median	Mean	Median	Mean	Median
ERPC	18.962	22.073	2.344	2.434	1.856	1.910
GERPC	20.812	23.852	2.556	2.640	1.938	1.995
LOMPC	21.139	24.000	2.553	2.648	1.928	1.974
KOMPC	21.701	24.000	2.562	2.650	1.932	1.978
GOMPC ($3rd$)	-	-	2.598	2.674	1.947	1.997
GOMPC ($4th$)	-	-	-	-	1.949	1.997

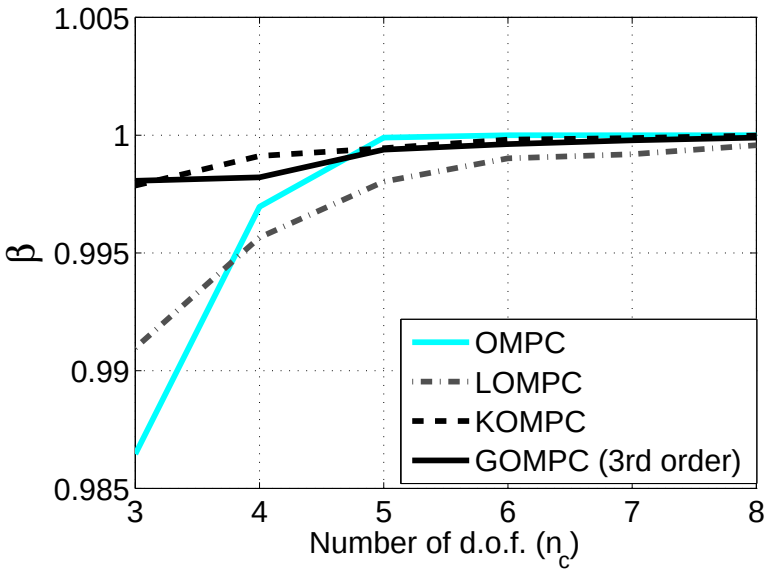

Figure 6.9: Comparison of average performance for Example 2.

Example 5 – $x \in \mathbb{R}^3$

Consider 500 single input single output $3rd$ order (i.e. $x \in \mathbb{R}^3$) random systems subject to input and state constraints

$$-1 \leq u_k \leq 1; \quad -\begin{bmatrix} 4 \\ 1.5 \\ 2 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 4 \\ 1.5 \\ 2 \end{bmatrix}. \tag{6.4}$$

Table 6.3: Percentages comparison of MCAS volumes with GERPC for all random systems.

Algorithm	Example 4			Example 5			Example 6		
	Greater	Equal	Less	Greater	Equal	Less	Greater	Equal	Less
	than/to GERPC								
LOMPC	40.4%	40.4%	19.2%	56.8%	23.4%	19.8%	21.2%	11.2%	67.6%
KOMPC	44.4%	42.2 %	13.4%	58%	24.2%	17.8%	25.2%	13.2%	61.6%
GOMPC ($3rd$)	-	-	-	61.6%	27.4%	11%	36.6%	31.6%	31.8%
GOMPC ($4th$)	-	-	-	-	-	-	41 %	27.2%	31.8%

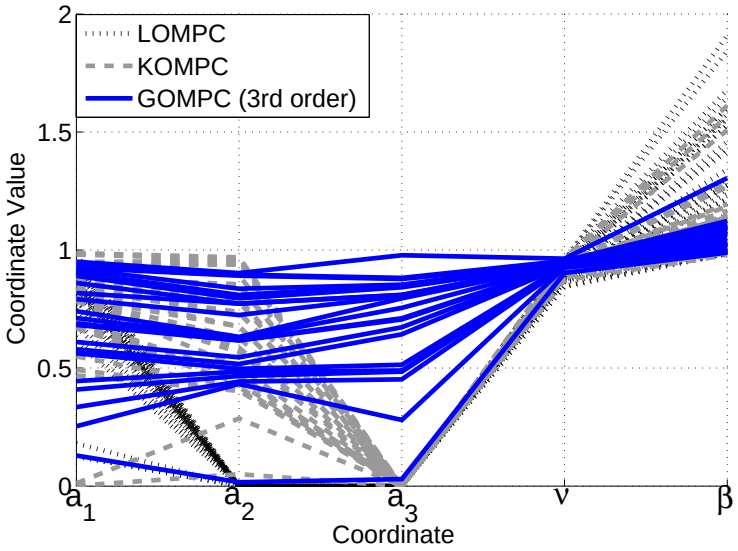

Figure 6.10: Comparison of parallel coordinates for Example 3.

The tuning parameters are $Q = C^T C$, $R = 1$, $n_c = 3$ and $\gamma = \infty$. For this example, GOMPC ($3rd$ order), Kautz and Laguerre dynamics are used as alternative parameterisation dynamics and tuned using pragmatic selection. The region of attractions are computed by selecting 200 state directions and computing, relatively, how far out in these directions a feasible solution exists.

Figure 6.16 shows the histograms of the MCAS volume for GOMPC ($3rd$ order), KOMPC, LOMPC, GERPC and ERPC. The histogram suggests that in many cases GOMPC ($3rd$ order), KOMPC and LOMPC have a larger MCAS volume than GERPC. Table 6.2 shows some statistics regarding the simulations. As shown, GOMPC ($3rd$ order) has a larger MCAS volume on average than KOMPC, LOMPC, GERPC and ERPC. The box and whisker plots in Figure 6.17 also shown that GOMPC ($3rd$ order) enlarges the region of attraction as the minimum value of MCAS volume is greater than KOMPC, LOMPC, GERPC and ERPC. The outliers suggests that there are some random systems where all algorithms may have a conservative region of attraction.

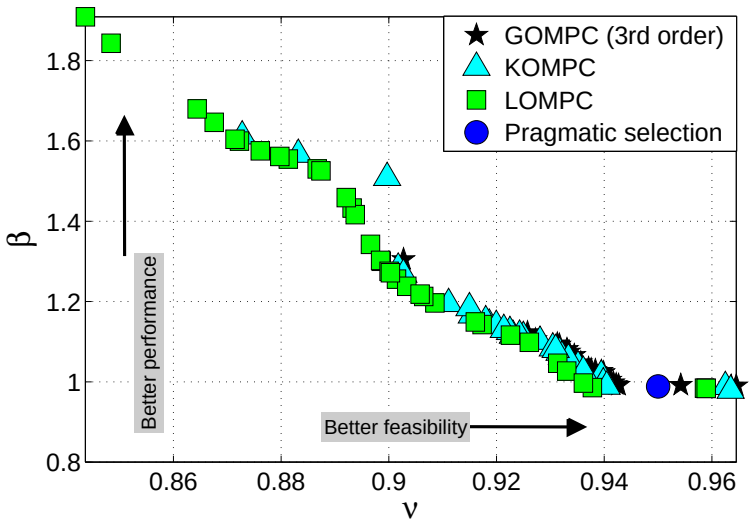

Figure 6.11: Comparison of pareto front for Example 3.

Example 6 – $x \in \mathbb{R}^4$

Consider 500 single input single output $4th$ order (i.e. $x \in \mathbb{R}^4$) random systems subject to input and state constraints

$$-1 \leq u_k \leq 1; \quad -\begin{bmatrix} 4 \\ 1.5 \\ 2 \\ 1 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 4 \\ 1.5 \\ 2 \\ 1 \end{bmatrix}. \tag{6.5}$$

The tuning parameters are $Q = C^T C$, $R = 1$, $n_c = 4$ and $\gamma = \infty$. For this example, GOMPC ($4th$ order), GOMPC ($3rd$ order), Kautz and Laguerre dynamics are used as alternative paremetrisation dynamics and tuned using pragmatic selection. The region of attractions are computed by selecting 500 state directions and computing, relatively, how far out in these directions a feasible solution exists.

Figure 6.18 shows the histograms of the MCAS volume for GOMPC ($4th$ order), GOMPC ($3rd$ order), KOMPC, LOMPC, GERPC and ERPC. The histogram suggests that in many cases GOMPC ($4th$ order) and GOMPC ($3rd$ order) has a larger MCAS volume than GERPC. However, GERPC enlarges the region of attraction compare to KOMPC and LOMPC. Table 6.2 shows some statistics regarding the simulations. As shown, GOMPC ($4th$ order) has a larger MCAS volume on average than GOMPC ($3rd$ order), KOMPC, LOMPC, GERPC and ERPC. The box and whisker plots in Figure 6.19 also

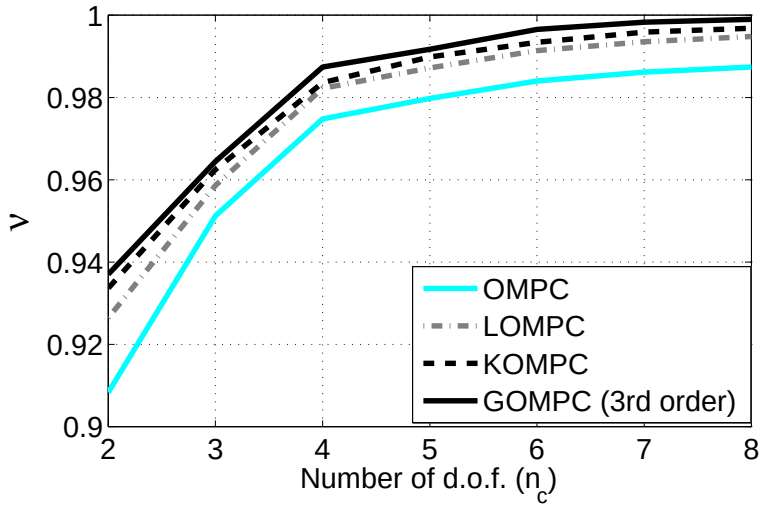

Figure 6.12: Comparison of average region of attraction volume for Example 3.

show that GOMPC ($4th$ order) enlarges the region of attraction as the minimum value of MCAS volume is greater than GOMPC ($3rd$ order), KOMPC, LOMPC, GERPC and ERPC. Similarly as seen in Example 5, the outliers suggests that there are some random systems where all algorithms may have a conservative region of attraction.

6.3.9 Mann-Whitney U test

The MCAS volume for GOMPC, KOMPC, LOMPC and GERPC are further analysed using two sample hypothesis tests. Figure 6.14, 6.16 and 6.18 show that the histogram of the MCAS volume is representing a non-normally distributed data. Therefore the non-parametric test can be used to test the null hypothesis (i.e. have the same median) between alternative algorithms (i.e. GOMPC, KOMPC and LOMPC) and GERPC or alternatively, whether the MCAS volume of alternative algorithms tend to be larger than GERPC [205]. The Mann-Whitney U test (also called the Wilcoxon rank-sum test) [206] is used and results are shown in Table 6.4.

Table 6.4 shows the statistical significance level using a P-value; this is the probability that the null hypothesis is true. The null hypothesis is rejected when significance level is less than 5%. The rejection of hypothesis is represented using h-value. If $h = 0$, it indicates the failure to reject the null hypothesis, whereas $h = 1$ indicates that the result would be highly unlikely under the null hypothesis.

It is shown that in Example 4, LOMPC and KOMPC enlarge the region of attraction

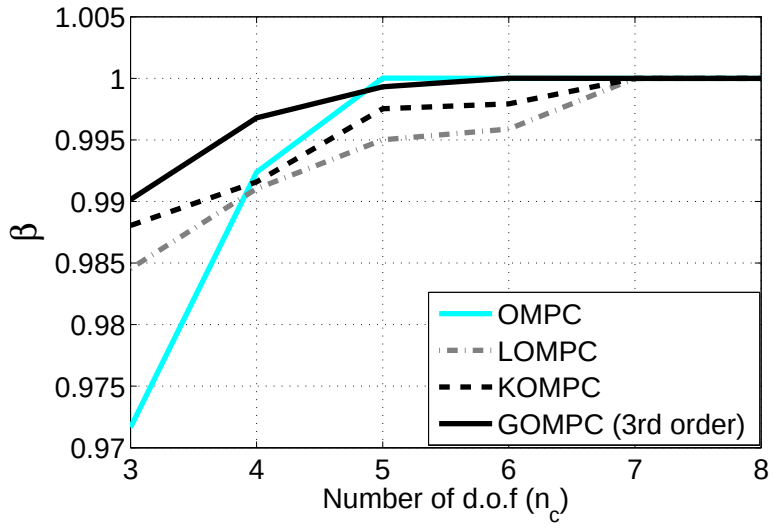

Figure 6.13: Comparison of average performance for Example 3.

as compared to GERPC as for both algorithms the P-value is less than 5% and $h = 1$. KOMPC has significantly enlarged the region of attraction as significant level is less than 1%, whereas LOMPC has 2% significant level. In Example 5, the significance level test shows that both GOMPC ($3rd$ order) and KOMPC enlarge the region of attraction compared to GERPC as the P-values are less than 5% and $h = 1$. However, LOMPC fail to reject the null hypothesis as significance level is 15% and $h = 0$. In Example 6, the significance level test indicates that for both GOMPC ($4th$ order) and GOMPC ($3rd$ order) fail to reject the null hypothesis test with 72% and 99% significance level. However, KOMPC and LOMPC reject the null hypothesis test and as shown in Figure 6.18 that GERPC has a larger MCAS volume. It is also shown in Table 6.2 that there is not much difference between the MCAS median using GOMPC and GERPC.

The significance test indicates that GOMPC, KOMPC and LOMPC enlarge the region of attraction as compared to GERPC using pragmatic parameter(s) selection. For higher dimensional system, it is interesting to observe that there is not much difference between the MCAS median using GOMPC and GERPC also shown in Table 6.2.

6.3.10 Summary

Table 6.3 shows the percentage comparison of the MCAS volume of random system with GERPC algorithm. It is shown that in Example 4, for 80.8 % random systems LOMPC has an equal or larger MCAS volume than GERPC, whereas for 19.2 % random

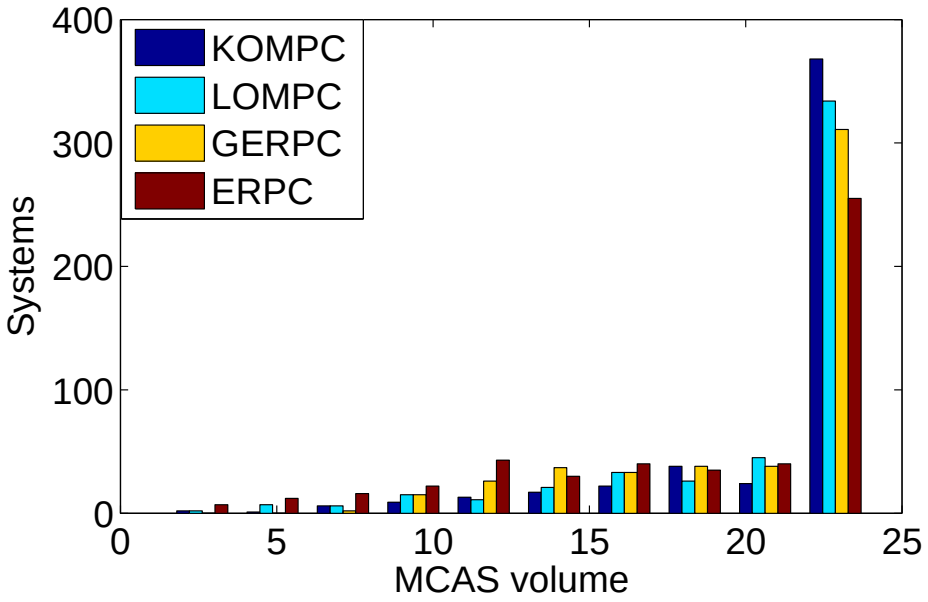

Figure 6.14: Histogram comparison of MCAS volume for Example 4.

systems LOMPC has a more conservative volume than GERPC. Similarly for KOMPC, for 86.6% random systems it gives an equal or larger MCAS volume than GERPC and there are 13.4% random system for which GERPC has a larger MCAS volume. In Example 5, LOMPC has 80.2% random systems, KOMPC has 82.2% random systems and GOMPC ($3rd$ order) has 89% random systems with equal or larger MCAS volume than GERPC. Whereas, there are 19.8% random systems for LOMPC, 17.8% random systems for KOMPC and 11 % random systems for GOMPC ($3rd$ order) for which GERPC has a larger MCAS volume. Similarly in Example 6, for LOMPC has 32.4% random systems, KOMPC has 38.4 % random systems, GOMPC ($3rd$ order) has 68.2 % random systems, GOMPC ($4th$ order) has 68.2 % random systems with equal or larger MCAS volume

Table 6.4: Mann-Whitney U test of MCAS volume between alternative algorithms and GERPC for random systems.

Algorithm	Example 4		Example 5		Example 6	
	P	h	P	h	P	h
	with GERPC					
LOMPC	0.0203	1	0.1482	0	1.89×10^{-8}	1
KOMPC	8.12×10^{-5}	1	0.0331	1	0.0031	1
GOMPC ($3rd$)	-	-	1.89×10^{-8}	1	0.9906	0
GOMPC ($4th$)	-	-	-	-	0.7198	0

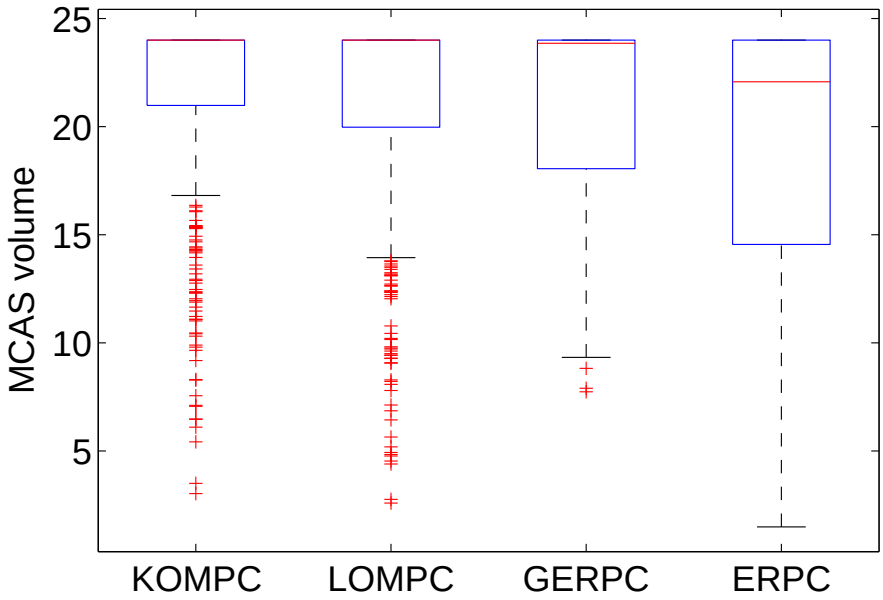

Figure 6.15: Box and whisker comparison of MCAS volume for Example 4.

than GERPC. There are 67.8% random systems for LOMPC, 61.6% random systems for KOMPC, 31.8% random systems for GOMPC ($3rd$ order), 31.8% random systems for GOMPC ($4th$ order) for which GERPC has a larger MCAS volume.

The statistical analysis shown in Table 6.2, Figure 6.14, 6.16 and 6.18 demonstrate that in many cases alternative parameterisations enlarge the region of attraction using pragmatic selection. From Table 6.3 and the outliers in Figure 6.15, 6.17 and 6.19 suggest that for some randomly selected system GERPC may be a better choice than alternative algorithms (i.e. LOMPC, KOMPC and GOMPC) using pragmatic selection. This is expected as the pragmatic choice is a suboptimal choice which provides a good starting point to tune the parameterisation dynamics.

6.4 Conclusion

The main contribution of this chapter is to extend the alternative parameterisation techniques available for shaping predicted input trajectories within predictive control and hence to give a more general class. The chapter then proposes a systematic mechanism to select the best parameterisation dynamics from this class. It is shown through numerical examples that alternative parameterisations using Laguerre, Kautz and higher

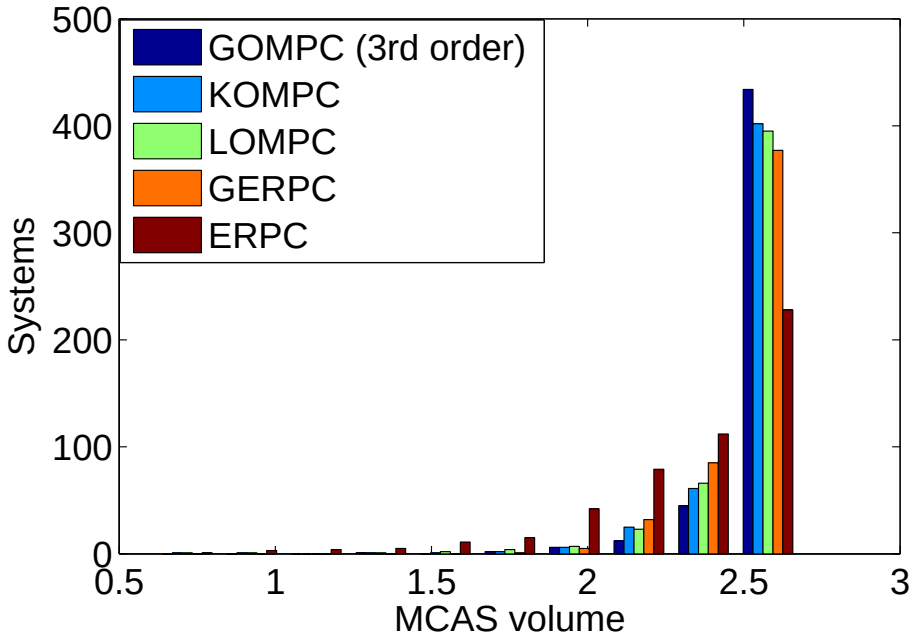

Figure 6.16: Histogram comparison of MCAS volume for Example 5.

order orthonormal functions can give significant feasibility benefits without a significantly degradation of closed loop performance and while facilitating lower dimensional optimisations than possible with a standard OMPC approach. While this benefit cannot be proven generically and for some cases is small, there is sufficient evidence to encourage users to try this out as, at times, the benefits can be very significant.

Two techniques for selecting the parameterisation dynamics were discussed based on optimal selection and a pragmatic approach based on stable closed loop poles. It is relatively straightforward to form an optimal selection procedure using a multi-objective optimisation based on trade off curves between MCAS volumes, performance and numbers of d.o.f.. Examples demonstrate that proposed trade off curves are clear and give good insight to the choices available; naturally, the definition of 'best' is somewhat subjective. Where such an offline analysis is not realistic, a pragmatic and simple selection method was demonstrated, by examples, to be highly effective in many cases.

The next chapter considers the computational efficiency of the generalised function parameterisation for OMPC. The computational efficiency is considered using both explicit and implicit solutions.

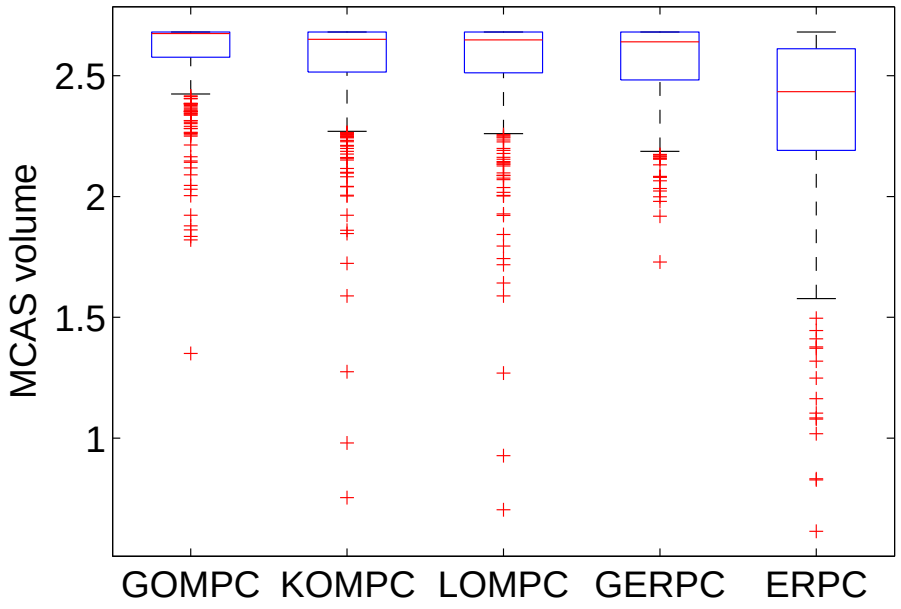

Figure 6.17: Box and whisker comparison of MCAS volume for Example 5.

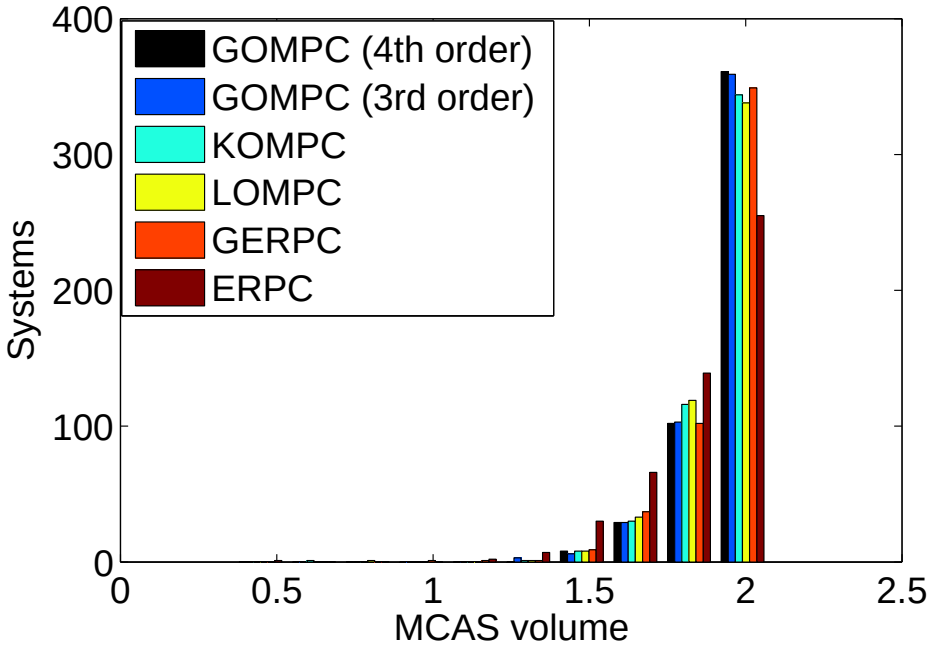

Figure 6.18: Histogram comparison of MCAS volume for Example 6.

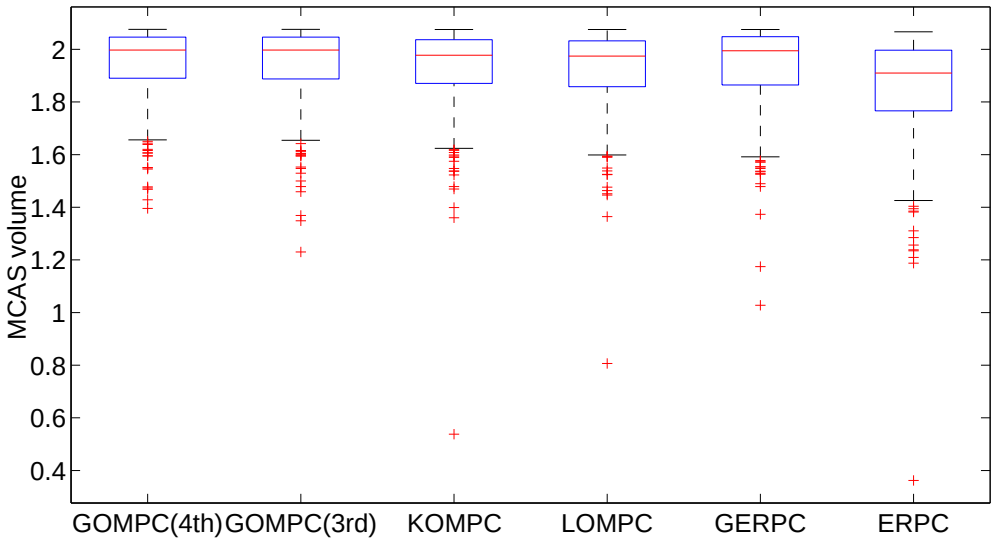

Figure 6.19: Box and whisker comparison of MCAS volume for Example 6.

Chapter 7

Computational analysis of generalised Optimal MPC

This chapter presents an **original contribution** to the thesis. The aim of this chapter is to consider the computational efficiency of using generalised function parameterisation for implicit and explicit solutions to MPC. The generalised function parameterisation facilitates substantial enlargement in the region of attraction with little or no detriment to performance while not increasing the number of optimisation variables. However, it is shown that some of the structure of the optimisation problem is lost when using an alternative parameterisation. This chapter considers the dense problem structure rising from removing any redundant constraints and then considers the online computational efficiency using the minimal sets.

Multi-parametric quadratic programming (mp-QP) is an alternative means of implementing conventional predictive control algorithms whereby one transfers much of the computational load to offline calculations. However, coding and implementation of this solution may be more burdensome than simply solving the original QP. This chapter also shows how generalised function parametrisations can be used in conjunction with mp-QP to achieve a large decrease in both the online computations and data storage requirements while increasing the region of attraction of the optimisation problem. Extensive simulation results which suggest there can still be benefits from using generalised function parameterisation.

This chapter is organised as follows: Section 7.1 presents the introduction and motivation of the chapter; Section 7.2 presents the necessary background about multi-parametric

programming. Section 7.3 proposed multiparametric QP based algorithm for GOMPC (mpGOMPC). Section 7.4 proposed active set method based algorithms for GOMPC with computational analysis. Simulation results are discussed in Section 7.5 and 7.6, showing the efficacy of the proposed algorithms and this is followed by conclusions in Section 7.7.

7.1 Introduction

Explicit solutions for the constrained MPC problem formulation [116] significantly increase the potential application areas for MPC. Explicit solutions to MPC problems are not intended to replace traditional implicit MPC, but rather to extend its area of applicability. MPC functionality can, with this, be applied to applications with low embedded hardware and with fast sampling rates [126, 207]. Software complexity and reliability is also improved, allowing the approach to be used in safety-critical applications.

The basic idea of the explicit solution is to solve, offline, all possible QP problems that can arise online. Within certain regions, the optimum predicted input trajectory has a known affine dependence on the state; mp-QP finds all possible active sets and the associated regions and affine dependence. The online optimisation can then replaced by set membership tests; if the state is inside a particular region, the control law is made of the associated control trajectory. However, although mp-QP is transparent, it may not reduce either coding complexity or computational effort as the number of computed regions, and hence, data storage, may grow exponentially in the prediction horizon [116]. Thus mp-QP could be unsuitable for large dimensional problems, or indeed any problem requiring a large number of regions.

Alternatively implicit solutions to MPC may be used with a collection of methods to greatly speed up the computation of the control action in MPC, using online optimisation. A number of researchers aimed to achieve a speed up online optimisation through the development of customized optimisation algorithms. These algorithms are exploiting the particular structure of the MPC problem to speed up the online computations [207]. The scenario considered in this chapter is one in which a QP problem has been solved by an active set method at the previous step and thus one only needs to solve a slightly perturbed QP from the preceding QP at the current time instant. In this case, active set methods are effective and the number of iterations required is typically a small polynomial dimension [151]. For active set methods, the initial seed is computed either solving Linear programming or the big M method [151].

The aim of this chapter is to consider the computational efficiency of generalised function

parameterisations within a predictive control algorithm using both explicit and implicit solutions. Conceptually the main question is whether changing the structure may improve the computational efficiency. Firstly to reduce the number of regions in the explicit MPC using generalised function parameterisations and therefore reduce the online computational burden (as this correlates to the number of online set-membership tests). A lot of the literature discusses how to reduce the number of on-line set-membership tests but there is little has appeared in the literature which reformulates the underlying optimisation problem in order to give significant reductions in complexity. In [208] the authors reduce data storage requirements by using an evaluation of a value function for the set-membership test, but the number of regions is not reduced. In [119] an efficient binary search tree introduced, but the offline computation of this can be prohibitive for complex controller partitions and the storage requirement may even increase. Other groups reduce the number of regions by allowing some suboptimality [117, 209] either in the performance index or the terminal region, although preliminary results are as yet unconvincing. Another alternative is to specify regions as hypercubes [133] to allow for efficient online search algorithms; however, as the structure of the controller is user-defined, it may not cover the entire controllable set. In [16] the authors interpolate two different laws achieving a large decrease in the number of regions but degrading the performance. This chapter develops a recent contribution [138] to this problem which used Laguerre MPC by proposing an alternative parameterisation of the d.o.f. in order to reduce the necessary online computations of optimal MPC. Specifically, the aim here is to extend this to generalised function parameterisations to consider how one can reduce the data storage requirements and the online implementation time for the associated mp-QP solution. The proposed procedure is based on Kautz and generalised function parameterisation in Section 4.4 and 5.3 of the d.o.f. which enlarges the region of attraction and without too much degradation in the performance of the closed-loop system; these procedures will henceforth be referred to as multi-parametric KOMPC (mpKOMPC) and multi-parametric GOMPC (mpGOMPC) respectively.

Secondly the focus of this chapter is to explore the implicit computational efficiency of generalised function parameterisation within an optimal MPC, using both a generic optimiser and an active set method. The computational efficiency using online optimisation may be obtained by using a warm start strategy and exploiting the structure of the QPs that arise in the MPC problem formulation. Nevertheless, one key question was still left unanswered: what is the computational efficiency using the online optimisation as the reduction in the number of d.o.f. may be compromised by a loss of structure in the optimisation? It is shown that some of the structure of the optimisation problem is lost when using alternative parameterisation. In contrast, Optimal MPC (OMPC) has strong structure which can be exploited in the active set method allowing relatively inexpensive

optimisation with a large number of d.o.f.. The aim is to consider a more holistic picture which allows the reduction of data storage requirements and the online computation time by removing any redundant constraints. The proposed procedure is based on standard OMPC, LOMPC, KOMPC and GOMPC.

7.2 Background

This section will introduce the background information on multi-parametric quadratic programming.

7.2.1 Multi-parametric Quadratic Programming (mp-QP)

Explicit solutions to constrained linear MPC problems can be obtained by solving a mp-QP where the parameters are the components of the state vector. In this thesis, the solutions to mp-QP problems are obtained using the Multi-Parameteric Toolbox [123]. In this section, the basic multi-parametric programming approach is summarised and for an in-depth discussion of multi-parametric programming the reader is referred to [116, 118, 119, 208, 210].

Background on mp-QP

Consider the following quadratic programming problem

$$\begin{aligned} J^*(x) = arg \min_u \; & u^T S_G u + x^T W_1 u \\ s.t. \; & Mx + Nu \leq d \\ & S_G > 0 \end{aligned} \tag{7.1}$$

where $u \in \mathbb{R}^{n_u}$ is the optimisation variable, $x \in \mathbb{R}^{n_x}$ is the parameter, with $N \in \mathbb{R}^{q \times n_u}$, $d \in \mathbb{R}^q$ and $M \in \mathbb{R}^{q \times n_x}$. In mp-QP, the main objective is to obtain the optimum value u^* for the whole range of parameters x, i.e. to obtain $u^*(x)$ as an explicit function of the parameter x.

The optimisation problem (7.1) can by some algebraic manipulation be reformulated as

$$\begin{aligned} J^*(x) = arg \ \min_z \ & z^T S_G z \\ s.t. \ & \widehat{G}x + Nz \leq d \end{aligned} \tag{7.2}$$

where $z = u + S_G^{-1} W_1^T x$ and $\widehat{G} = M - N S_G^{-1} W_1^T$. A mp-QP consists of the following steps [118]

Active constraint identification

A feasible $\hat{x}$ is determined and the associated QP (7.1) is solved. This will yield the optimiser z and active constraints $\mathcal{A}(\hat{x})$ defined by the inequalities that are active at solution, i.e.

$$\mathcal{A}(\hat{x}) = \{i \in \{i, 2, \ldots, q\} \mid \widehat{G}_{\{i\}}\hat{x} + N_{\{i\}}z - d_{\{i\}} = 0\} \tag{7.3}$$

where $\widehat{G}_{\{i\}}$, $N_{\{i\}}$ and $d_{\{i\}}$ denotes the ith row of the matrices $\widehat{G}$, N and d respectively of q number of constraints. The active constraints form the matrices $\widehat{G}_{\mathcal{A}}$, $N_{\mathcal{A}}$ and $d_{\mathcal{A}}$.

Region computation

As shown in [118], the mp-QP problem (7.2) can be solved by applying the Karush-Kuhn-Tucker (KKT) conditions to obtain an explicit representation of the optimum value $u^*(\hat{x})$ which is valid in some neighbourhood of $\hat{x}$.

$$S_G z + N^T \lambda = 0 \tag{7.4}$$

$$\lambda^T (Nz + \widehat{G}\hat{x} - d) = 0 \tag{7.5}$$

$$\lambda \geq 0 \tag{7.6}$$

$$\widehat{G}\hat{x} + Nz \leq d \tag{7.7}$$

The optimised variable z^* can be solved from

$$z^* = -S_G^{-1} N^T \lambda. \tag{7.8}$$

Constraints in condition (7.5) can be separated into active and inactive constraints. For inactive constraints the Lagrange multipliers $\lambda = 0$, whereas for active constraints $\lambda > 0$

and inequality constraints are changed to equality constraints. Substituting for z from (7.8) into the equality constraints gives

$$-N_{\mathcal{A}} S_G^{-1} N_{\mathcal{A}}^T \lambda - d_{\mathcal{A}} + \widehat{G}_{\mathcal{A}} \hat{x} = 0, \quad \lambda > 0 \tag{7.9}$$

and yields expressions for the active Lagrange multipliers

$$\lambda = -(N_{\mathcal{A}} S_G^{-1} N_{\mathcal{A}}^T)^{-1} (d_{\mathcal{A}} - \widehat{G}_{\mathcal{A}} \hat{x}). \tag{7.10}$$

The optimum value z^* and optimal control trajectory u^* are thus given as affine functions of $\hat{x}$

$$\begin{aligned} z^*(\hat{x}) =& S_G^{-1} N_{\mathcal{A}}^T (N_{\mathcal{A}} S_G^{-1} N_{\mathcal{A}}^T)^{-1} (d_{\mathcal{A}} - \widehat{G}_{\mathcal{A}} \hat{x}) && (7.11) \\ u^*(\hat{x}) =& z^*(\hat{x}) - S_G^{-1} W_1^T \hat{x} \\ =& S_G^{-1} N_{\mathcal{A}}^T (N_{\mathcal{A}} S_G^{-1} N_{\mathcal{A}}^T)^{-1} (d_{\mathcal{A}} - \widehat{G}_{\mathcal{A}} \hat{x}) - S_G^{-1} W_1^T \hat{x} \\ =& \hat{K}_r \hat{x} + t_r && (7.12) \end{aligned}$$

where

$$\begin{aligned} \hat{K}_r =& - S_G^{-1} N_{\mathcal{A}}^T (N_{\mathcal{A}} S_G^{-1} N_{\mathcal{A}}^T)^{-1} \widehat{G}_{\mathcal{A}} - S_G^{-1} W_1^T && (7.13) \\ t_r =& S_G^{-1} N_{\mathcal{A}}^T (N_{\mathcal{A}} S_G^{-1} N_{\mathcal{A}}^T)^{-1} d_{\mathcal{A}}. && (7.14) \end{aligned}$$

In the next step, the set of states is determined where the optimiser $u^*(\hat{x})$ satisfies the same active constraints and is optimal. Specifically the control regions $\mathcal{P}_r = \{x \in \mathbb{R}^{n_x} \mid \widehat{M}_r x \leq \widehat{d}_r\}$ is computed for

$$\widehat{M}_r = \begin{bmatrix} N(\hat{K}_r + S_G W_1^T) - \widehat{G} \\ (N_{\mathcal{A}} S_G^{-1} N_{\mathcal{A}}^T)^{-1} \widehat{G}_{\mathcal{A}} \end{bmatrix} \tag{7.15}$$

$$\widehat{d}_r = \begin{bmatrix} d - \widehat{G} t_r \\ -(N_{\mathcal{A}} S_G^{-1} N_{\mathcal{A}}^T)^{-1} d_{\mathcal{A}} \end{bmatrix}. \tag{7.16}$$

State-space exploration

Once the controller region is computed, the algorithm proceeds iteratively until the entire feasible state $\mathcal{X}_{n_u}$ is covered with controller regions $\mathcal{P}_r$, i.e. $\mathcal{X}_{n_u} = \bigcup_{r=1}^{r_n} \mathcal{P}_r$.

7.3 Generalised OMPC solved by mp-QP or mpGOMPC

The solution of the optimisation problem using mp-QP can be applied to KOMPC and GOMPC as they all take the form of standard quadratic programming; this section summarises the key points for completeness.

If the optimisation problem is solved parametrically as an explicit function of the initial conditions x_0, the optimal feedback law $u = f(x_0)$ takes a form of a lookup table. The online optimisation of such table then reduces to a simple set membership test, also known as the point location problem. Here, the table has to be searched through and the element which contains the current state measurement has to be found.

It is obvious that the KOMPC cost function (4.21) and GOMPC cost function (5.12) are in the form of (7.1), with $W_1 = 0$ and therefore $z = u$. The algorithm based on mp-QP for GOMPC is presented in Algorithm 7.1.

Algorithm 7.1. *mpGOMPC*

Off-line

1. *Select the order of prediction dynamics, that is the number of poles a_i in A_G (5.5).*

2. *Select specific values for the poles a_i.*

3. *Determine the predicted cost, in terms of perturbations c_k is*

$$J_k = \sum_{i=0}^{\infty} c_{k+i}^T S c_{k+i}. \tag{7.17}$$

Substitute in from (5.5) *and* (5.11) *the GOMPC predictions of $c_{k+i} = G_i^T \underrightarrow{\rho}_k$ to give*

$$J_{G,k} = \sum_{i=0}^{\infty} \underrightarrow{\rho}_k^T G_i S G_i^T \underrightarrow{\rho}_k. \tag{7.18}$$

Finally, substitute $G_i = A_G G_{i-1}$ and hence:

$$J_{G,k} = \underrightarrow{\rho}_k^T \left[\sum_{i=0}^{\infty} A_G^i G_0 S G_0^T (A_G^i)^T\right] \underrightarrow{\rho}_k = \underrightarrow{\rho}_k^T S_G \underrightarrow{\rho}_k. \tag{7.19}$$

4. *Define the constraint inequalities associated to* (4.3) *in the form*

$$Mx_k + NH_G \underset{\rightarrow}{\rho}_k \leq d. \tag{7.20}$$

5. *Solve parametrically the following QP.*

$$\begin{aligned} \rho_k = arg \min_{\rho_k} \quad & J_{G,k} \\ s.t. \quad & Mx_k + NH_G \, \rho_k \leq d \end{aligned} \tag{7.21}$$

6. *Store the optimal predicted input trajectories* ρ_k *(which implies* $C_G = G^T \rho_k$*) and associated regions.*

On-line

1. *Find the corresponding solution* (C_G) *of the optimisation problem* (7.21) *associated to the current state* x_k.
2. *Implement the control law* $u_k = -Kx_k + e_1^T C_G$, *where* e_1 *is the standard basis vector.*

Remark 7.3.1. Note that the procedures in [119, 208] may be used in combination with mpGOMPC to obtain even greater reductions in complexity.

7.4 Generalised function parameterisation in MPC using active set methods

This section considers the optimisation for LOMPC, KOMPC and GOMPC in detail and in particular the potential, or not, to exploit any structure for an efficient active set method.

7.4.1 Optimisation Structure of GOMPC

The generalised function parameterisation approaches choose generic 'stable' basis vectors to enlarge the region of attraction. These approaches reduce the d.o.f. to overcome the

trade off between the region of attraction and performance. The active set method (ASM) requires a computational time which is cubic in d.o.f. to solve the dense formulation in (4.3) and (7.21). To obtain both good performance and a large MCAS, OMPC might require a large n_c (d.o.f.) which thus would compromise the computational efficiency of an ASM. Various methods can be used to speed up the computation, for example one strategy is to exploit the structure of the QP that arises in a simple MPC formulation which includes all the constraints [127], that is including the redundant ones. Of course this may require more storage space with a large n_c for computations.

OMPC is a special case of an alternative parameterisation as shown in Section 4.3 and 5.2. It has a lower shift dynamic matrix, whereas alternative parameterisations (i.e. Laguerre, Kautz and higher order orthonormal function parameterisation) have lower triangular dynamic matrices. This is the main difference which compromises the QP structure resulting from alternative parameterisations.

In generalised function parameterisation approaches the QPs become dense and the problem structure is less obvious, and thus a tailored ASM method is as yet not available with a consequent increase in computation operations per step for the same n_c as OMPC. In terms of online computation, OMPC requires more d.o.f. to improve the region of attraction and the question is, should we use a parameterised MPC with a low n_c or add more d.o.f. to a conventional OMPC approach? In this chapter focus will given to comparing the computational efficiency of the OMPC and parameterised algorithms. Critically however, as the formulation with GOMPC is already dense and there is no structural advantage in retaining redundant constraints and thus it makes sense to remove all the redundant constants before moving to the online computation load. This also reduces the storage requirement.

7.4.2 Computational complexity using online optimisation

In GOMPC 5.1, an offline computational burden required to perform an alternative parameterisation is expressed by giving the total number of floating point operations or flops. The number of flops count required is in terms of generating the prediction matrix H_G, NH_G and calculating the parameterised cost $H_G^T diag(S) H_G$. Using mth order prediction dynamics with n_ρ rows of H_G required $m^3(n_\rho - 1)$ flops. The prediction cost requires $2n_c^2 + n_c n_\rho$ and NH_G requires $(n_x + n_u)n_\rho^3$ flops. Hence generalised function parameterisations require $m^3(n_\rho - 1) + 2n_c^2 + n_c n_\rho + (n_x + n_u)n_\rho^3$ extra offline flops. The online computational complexity of OMPC algorithm using IPM is linear in the horizon and cubic in the state and input dimensions [145], [159]. However, all the proposed parameterisations will be cubic in the number of d.o.f., the state and input dimensions.

Table 7.1: Computational burden using IPM

Algorithm	Flops	
	Offline (extra)	Online
OMPC		$n_c + n_x^3 + n_u^3$
GOMPC	$m^3(n_\rho - 1) + 2n_c^2 + n_c n_\rho + (n_x + n_u)n_\rho^3$	$n_c^3 + n_x^3 + n_u^3$

Table 7.1 shows the computational burden using online optimisation. In terms of online implementation computation OMPC requires more d.o.f. to improve the region of attraction as compared to alternative parameterisation techniques (i.e. LOMPC, KOMPC and GOMPC). Alternative parameterisations have however give a systematic approach for handling feasibility/performance trade offs in general.

7.4.3 Active set method applied to GOMPC

An active set method (ASM) is introduced here to solve the optimisation problem in (7.21). An Active set method converts the proposed optimisation into a sequence of equalities problems (EPs), involving only equality constraints, which are solved to generate a sequence of iterates converging to the solution. At each iteration an active set method solves a Karush-Kuhn-Tucker (KKT) system defined by the 'proposed' active constraints. Consequently, when a significant number of constraints are active, the 'system' is much smaller in terms of constraints than in an interior point method.

The algorithm uses offline computations to remove the redundant constraints to reduce the online computational load. This is especially desirable since the constraint matrices are by necessity dense (because the structure of the optimisation is lost when using function parameterisation). This becomes advantageous as the problem size decreases and the overall online computational effort involved is comparable with that of lower dimensional problems. This chapter demonstrates the resulting improvements in both computational burden and size of stabilisable sets (or MCAS).

The following gives a brief description of an active set solver for the QP (7.21) and provides details of the computation involved in each step. Introducing the Lagrangian associated with the problem (7.21) can be defined as:

$$L(\rho, \lambda) = \frac{1}{2}\rho_k^T S_G \rho_k + \lambda^T (M x_k + \widehat{N}\rho_k - d) \tag{7.22}$$

where $\widehat{N} = NH_G$. The KKT conditions of (7.21) are given similarly as in (7.4),(7.5), (7.6) and (7.7)

$$\begin{aligned} S_G\rho_k + \widehat{N}^T\lambda = 0, \\ d - Mx_k - \widehat{N}\rho_k \geq 0, \\ (Mx_k + \widehat{N}\rho_k - d)^T\lambda = 0, \\ \lambda \geq 0. \end{aligned} \tag{7.23}$$

At each iteration, an active set method solves a KKT system (7.23) defined by the active constraints. The null space method is used to calculate the solution of the KKT system (7.23), the QR factorisation is introduced in order to calculate a null space basis matrix, which improves the computational efficiency. The procedure is summarised in algorithm 7.2 [211].

Algorithm 7.2. *GOMPC using ASM*

Off-line

1. *Set* $\widehat{As} = \begin{bmatrix} M & NH_G \end{bmatrix}$, $\widehat{ds} = d$ *and* $X = \begin{bmatrix} x \\ \rho \end{bmatrix}$. *The problem is to remove all redundant inequalities in* $\widehat{As}X \leq \widehat{ds}$ *to obtain an irredundant description description* $AsX \leq ds$ *with same set results.*

2. *Set* $i = 1$, $As = [\,]$ *and* $ds = [\,]$.

3. *If* $\max_X \widehat{As}_i X > \widehat{ds}_i$, *then set* $As = \begin{bmatrix} As \\ \widehat{As}_i \end{bmatrix}$ *and* $ds = \begin{bmatrix} ds \\ \widehat{ds}_i \end{bmatrix}$.

4. *If* $i <$ *length of* As *then set* $i = i + 1$ *and goto step 3, else terminate;*

5. *The irredundant description is given by* $AsX \leq ds$.
   The maximisation in step 3 is a linear programming (LP).
   $M = As_{(:,1:n_x)}$, $\widehat{N} = As_{(:,n_x+1:end)}$ *and* $d = ds$.

On-line tasks:

1. *Set* $\mathcal{W}_0$ *as set of active constraints. The set of active constraint* $\mathcal{W}_k$ *at iteration* k

for point x_k can be defined as

$$\mathcal{W}_k = \{Mx_k + \widehat{N}\rho_k = d\}. \tag{7.24}$$

2. *At first, check whether ρ_k is optimal in the subspace defined by $\mathcal{W}_k$ (initial seed is computed using linear programming).*
   Then, define a move direction $\hat{p}$ and express $J_{G,k}$ as a function of $\hat{p}$:

$$\begin{aligned} f(\rho_k + \hat{p}) &= \frac{1}{2}(\rho_k + \hat{p})^T S_G(\rho_k + \hat{p}) \\ &= \frac{1}{2}\hat{p}^T S_G \hat{p} + \hat{p}^T S_G \rho_k + const. \end{aligned}$$

Take a small step in direction $\hat{p}$ defined by the QP with equalities

$$\begin{aligned} \min_{\hat{p}} \ & \frac{1}{2}\hat{p}^T S_G \hat{p} + \hat{p}^T S_G \rho_k \\ s.t. \quad & \widehat{N}_k^T \hat{p} = 0 \end{aligned} \tag{7.25}$$

The solution of the sub-problem (7.21) *is given by the solution of the KKT system*

$$\begin{pmatrix} S_{G_k} & \widehat{N}_k^T \\ \widehat{N}_k & 0 \end{pmatrix} \begin{pmatrix} \hat{p} \\ \zeta \end{pmatrix} = \begin{pmatrix} -S_{G_k}\rho_k \\ 0 \end{pmatrix} \tag{7.26}$$

   (a) *If the solution of* (7.21) *is $\hat{p} = 0$ then ρ_k is optimal in the current subspace. Proceed to stage 3 below*

   (b) *Otherwise,*

$$\rho_{k+1} = \rho_k + \hat{\alpha}\hat{p}, \ 0 < \hat{\alpha} \leq 1. \tag{7.27}$$

   The step size $\hat{\alpha}$ must be chosen to maintain feasibility. So

$$\hat{\alpha} = min\left(1, \min_{n_i^T \hat{p} > 0} \frac{d_i - m_i^T x_j - n_i^T \rho_k}{n_i^T \hat{p}}\right) \tag{7.28}$$

   A constraint i which would yield $\hat{\alpha} < 1$ in (7.28) *is a blocking constraint. Add*

blocking constraint to the working set $\mathcal{W}_k$ to form $\mathcal{W}_{k+1}$ and update the iterate by (7.27)*. Otherwise, $\hat{\alpha} = 1$ in* (7.28)*, then working set, $\mathcal{W}_{k+1} = \mathcal{W}_k$.*

3. *An optimal ρ_k has been found for $\mathcal{W}_k$, check optimality: set $\zeta_i^k = 0 \; \forall \; i \notin \mathcal{W}_k$, the other Lagrange multipliers at this point are known from* (7.26)*.*

   (a) *If $\zeta \geq 0$ then all KKT conditions hold and the optimal point has been found.*

   (b) *Otherwise, there is a component $\zeta_q < 0$. Then $\mathcal{W}_{k+1}$ is formed by dropping q constraint from $\mathcal{W}_k$, and iteration repeated.*

7.5 Numerical Examples

The purpose of this section is to compare the explicit implementation (i.e. mp-QP solution) and active set methods for the alternative parameterisation algorithms (i.e. LOMPC, KOMPC and GOMPC) and OMPC algorithm. The prime interest is to compare three aspects: (i) the MCAS; (ii) the number of inequalities to describe MCAS; (iii) the complexity of the different algorithmic solutions (in essence the number of regions and computation time) so that some comments can be made about the potential implementation of GOMPC.

7.5.1 Simulation setup using multiparametric solution

The OMPC algorithm with $n_c = n_x$ is used as a basis for comparisons. All algorithms (OMPC, LOMPC, KOMPC and GOMPC) provide stability and feasibility properties but the region of the maximum controller admissible control admissible set (MCAS) for each of them varies. Hence, it is necessary to compare both the complexity of the algorithms and the region of the associated MCAS. The comparisons are based on 500 random systems with $x \in \mathbb{R}^2$, $x \in \mathbb{R}^3$, and $x \in \mathbb{R}^4$ (total of 1500 systems).

The $2nd$ order ($x \in \mathbb{R}^2$) random systems are subject to input and state constraints as:

$$-1 \leq u_k \leq 1, \quad -\begin{bmatrix} 4 \\ 1.5 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 4 \\ 1.5 \end{bmatrix}, \tag{7.29}$$

with the performance objective weighting matrices as $R = 1$ and $Q = diag(1, 0)$ for $n_c = 2$, and 200 state directions are chosen for the initial states.

For the $3rd$ order ($x \in \mathbb{R}^3$) random systems, the input and state constraints are:

$$-1 \leq u_k \leq 1, \quad -\begin{bmatrix} 4 \\ 1.5 \\ 2 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 4 \\ 1.5 \\ 2 \end{bmatrix}, \tag{7.30}$$

with the performance objective weighting matrices as $R = 1$ and $Q = diag(1, 0, 0)$ for $n_c = 3$ using 200 different state directions.

Similarly, the inputs and states constraints for $4th$ order ($x \in \mathbb{R}^4$) random systems are:

$$-1 \leq u_k \leq 1, \quad -\begin{bmatrix} 4 \\ 1.5 \\ 2 \\ 4 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 4 \\ 1.5 \\ 2 \\ 4 \end{bmatrix}, \tag{7.31}$$

with the performance objective weighting matrices as $R = 1$ and $Q = diag(1, 0, 0, 0)$ for $n_c = 4$ using 500 different state directions.

The Laguerre, Kautz and generalised function dynamics are tuned using the pragmatic selection.

7.5.2 Simulation setup Using Active Set Methods

A simple implementation of the active set method is developed in MATLAB, which handles the case of a quadratic objective and box constraints. The main purpose is to compare the timing results for the alternative parameterisation algorithms and standard OMPC algorithm using ASM and quadprog.m methods (using large-scale algorithm) on a 3.26 GHz Intel Core 2 Duo running Microsoft Window XP. The prime interest is to compare the complexity of the different algorithm solutions (in essence computation time).

The OMPC with $n_c = 3$ is used as a basis for comparisons. The comparison is based on 4 random systems with following dimensions $x \in \mathbb{R}^4$, $x \in \mathbb{R}^{10}$, $x \in \mathbb{R}^{16}$ and $x \in \mathbb{R}^{30}$. The inputs and states for all systems were constrained to $-0.08 \leq u \leq 0.08$ and $-2.4 \leq x_j \leq 2.4$, $(j = 4, 10, 16, 30)$ with performance objective weighting matrices $R = I$ and $Q = I$.

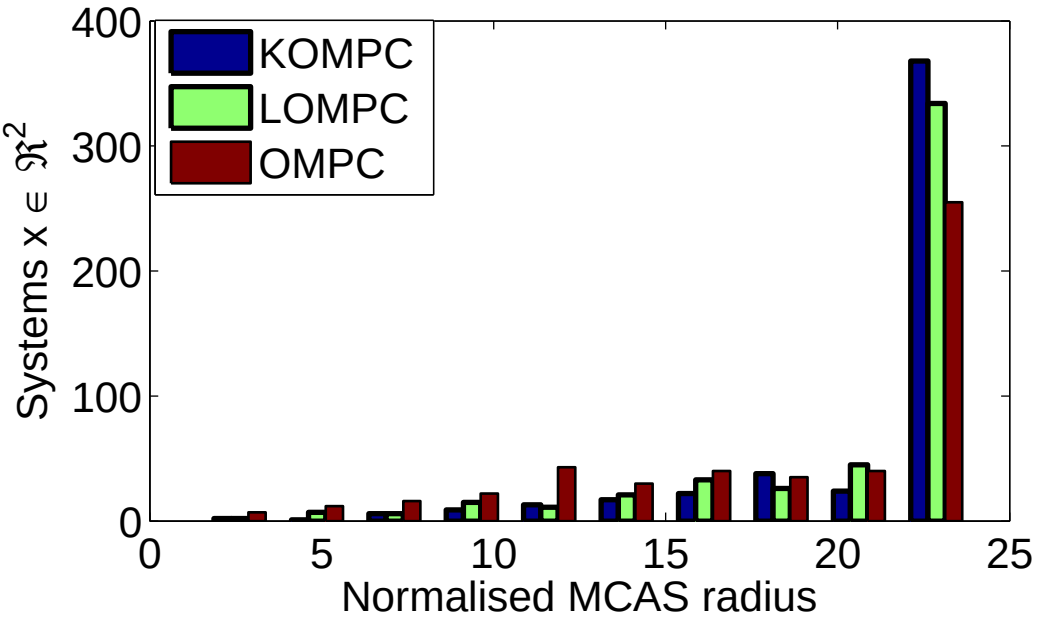

Figure 7.1: Comparison of the MCAS radius for 500 random systems ($x \in \mathbb{R}^2$)

7.5.3 MCAS comparisons with $n_c = n_x$

The MCAS is estimated using a large number of different directions in the state space. For each direction, the distance from the origin to the boundary of the MCAS is determined; the larger the distance, better the feasibility. Finally the mean is calculated for all state directions to represent the MCAS radius. Here the MCAS comparison is presented for completeness.

The MCAS radii are compared in Table 7.2, 7.3, 7.4 and Figure 7.1, 7.5, 7.8 for 500 random systems. In the figures, the x-axis serves as MCAS radius variations and the y-axis indicates the random systems. For $x \in \mathbb{R}^2$ using the statistical analysis in Table 7.2, the mean and minimum value shows that KOMPC results in larger regions of attraction than LOMPC and OMPC algorithms. Similarly from Table 7.3 and 7.4 for $3rd$ and $4th$ dimensional systems, GOMPC results in larger regions of attraction than KOMPC, LOMPC and OMPC algorithms. Alternative parameterisations noticeably results in larger regions of attraction than OMPC and it is clear that the generalised dynamics with $n_G = n_x$ has a larger MCAS than OMPC, and $n_G \leq n_c$ as already shown in the previous chapters.

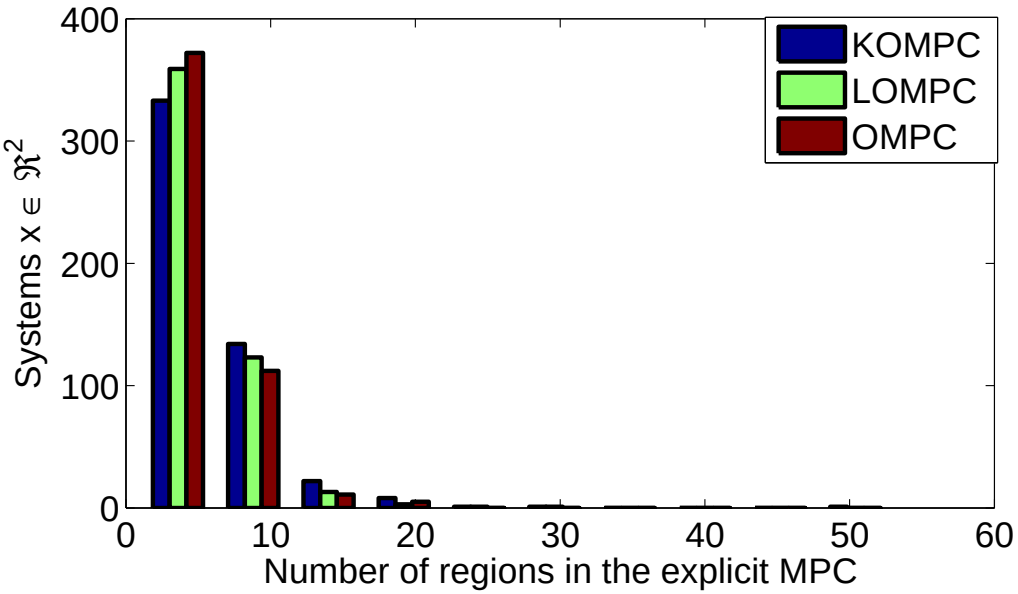

Figure 7.2: Comparison of number of regions for explicit MPC solution for 500 random systems ($x \in \mathbb{R}^2$)

7.5.4 Complexity comparisons using number of regions in the explicit MPC with $n_c = n_x$

The computational complexity is compared by finding all possible active set and the associated regions and control trajectories using a solution of the mp-QP. The online computational load for the set membership test is less than or equal to the total number of regions.

Figure 7.2, 7.6, 7.9 and Table 7.2, 7.3, 7.4 give a statistical comparison of the number of regions in the explicit MPC of mpOMPC, mpLOMPC, mpKOMPC and mpGOMPC versus 1500 random systems. It is noted that in many cases the average storage requirements of mpOMPC are less but with a smaller MCAS.

Table 7.2 shows the statistics for 2-dimensional random systems, on average KOMPC, LOMPC and OMPC requires 5 explicit MPC regions. In the worst case, KOMPC requires 53 and LOMPC requires 29 regions, whereas OMPC requires 21 regions. Figure 7.2 shows the histogram for the number of regions using the explicit MPC solution, it is seen that for OMPC most of occurrence is on the lower number of the regions. For second order dynamics, alternative function (LOMPC and KOMPC) parameterisation (in the worst case) requires a large number of regions for the explicit MPC solution as compared to OMPC.

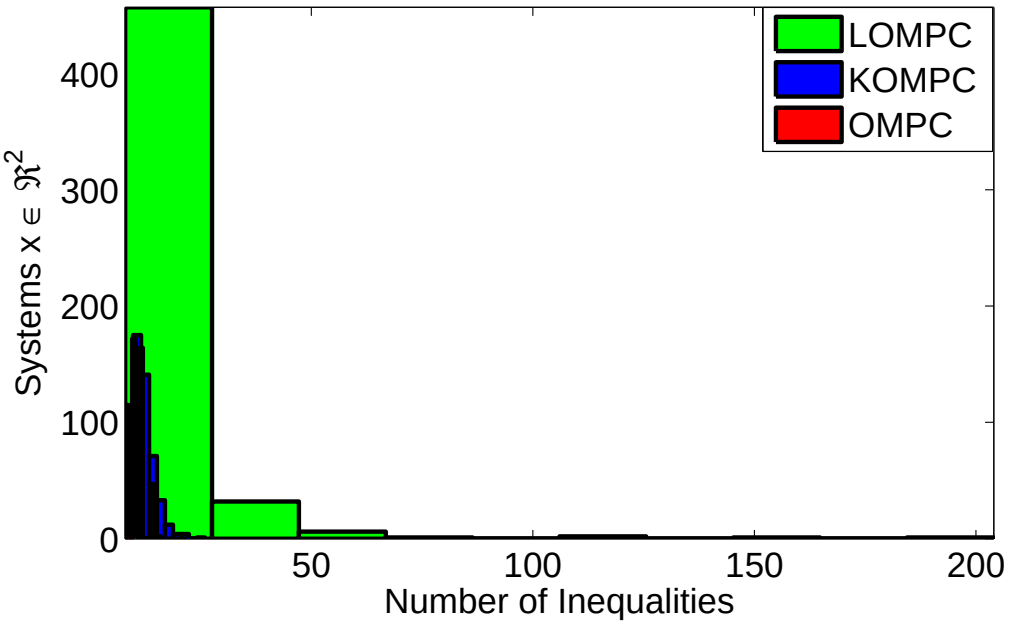

Figure 7.3: Comparison of number of inequalities for 500 random systems ($x \in \mathbb{R}^2$)

For 3-dimensional systems ($3rd$ order dynamics), the statistical analysis is shown in Table 7.2. It is shown that on average GOMPC requires 25 regions, KOMPC and LOMPC require 15 regions whereas, OMPC requires 14 regions. In the worst case, the alternative parameterisation requires a large number of regions than OMPC. The maximum number of regions required using GOMPC, KOMPC and LOMPC are 139, 131 and 99 respectively, whereas, OMPC required only 73 regions. Figure 7.6, shows the histogram of the number of regions using 500 random systems. It is noted that, KOMPC, LOMPC and OMPC have more occurrences on the lower number of regions than GOMPC.

In the case of the 4-dimensional systems, on average the GOMPC, KOMPC, LOMPC and OMPC requires 176, 75, 61 and 43 numbered of regions respectively whereas, the maximum number of regions requires are 1187, 1042, 1075 and 247 respectively as shown in Table 7.4. Similarly to $3rd$ order system, the use of alternative parameterisation typically increased the storage requirements of mp-QP partitions. The histogram in Figure 7.9 shows that OMPC has more occurrences with fewer regions compared to alternative parameterised algorithms. It is concluded that for a higher order dynamic system, the alternative parametrisation increases the number of the regions of the explicit MPC.

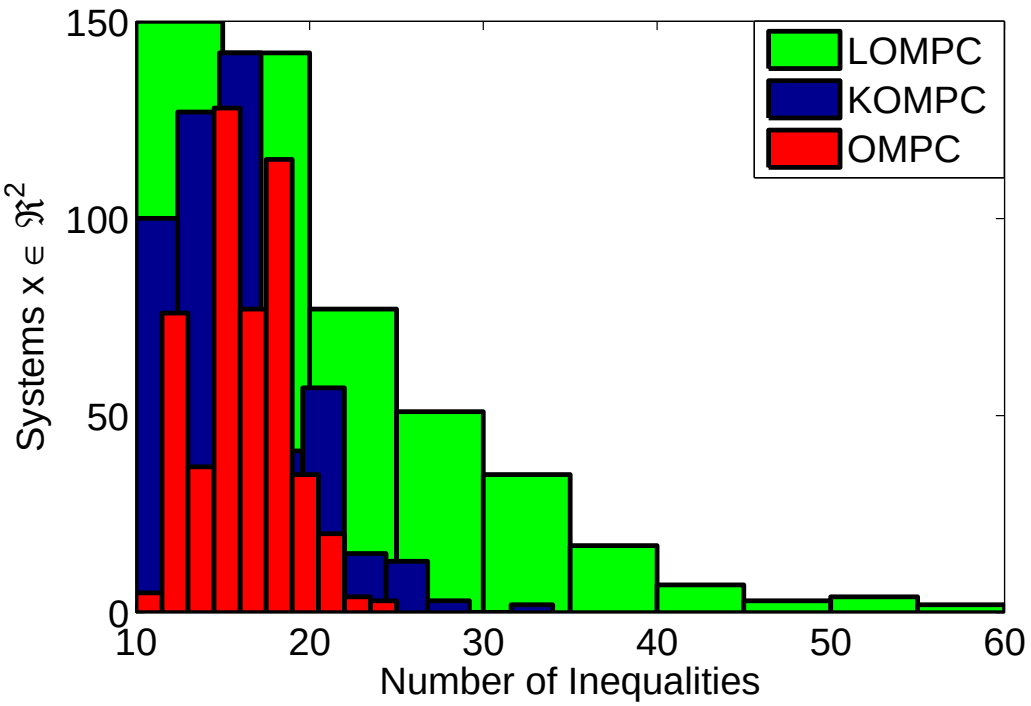

Figure 7.4: Comparison of number of inequalities for 500 random systems ($x \in \mathbb{R}^2$) **with inequalities** $\leq$ **60**

7.5.5 Number of inequalities to describe the MCAS

Figures 7.3, 7.4, 7.7 and 7.10 show the histogram comparison of the number of inequalities to describe the MCAS for 500 random systems. The statistics are shown in Table 7.2, 7.3 and 7.4 for GOMPC, KOMPC, LOMPC and OMPC algorithms. It is shown that on average parameterised algorithms increase the number of inequalities to describe the MCAS compared to OMPC, whereas OMPC has a smaller MCAS than parameterised algorithms.

For second order random systems using $n_c = 2$, on average KOMPC, LOMPC and OMPC require 16, 23, and 16 numbers of inequalities respectively whereas, the maximum number of inequalities are 25, 263 and 34 respectively as shown in Table 7.2. From Figure 7.3 and 7.4, the number of inequalities using KOMPC and OMPC is less than 50. However, the number of inequalities using LOMPC varies from 10 to 263, but there are only few random system for which the number of inequalities is more than 50. KOMPC slightly increases the number of inequalities and enlarges the region of attraction compared to OMPC. From statistical analysis, KOMPC is preferred in general over LOMPC for 2-dimensional systems.

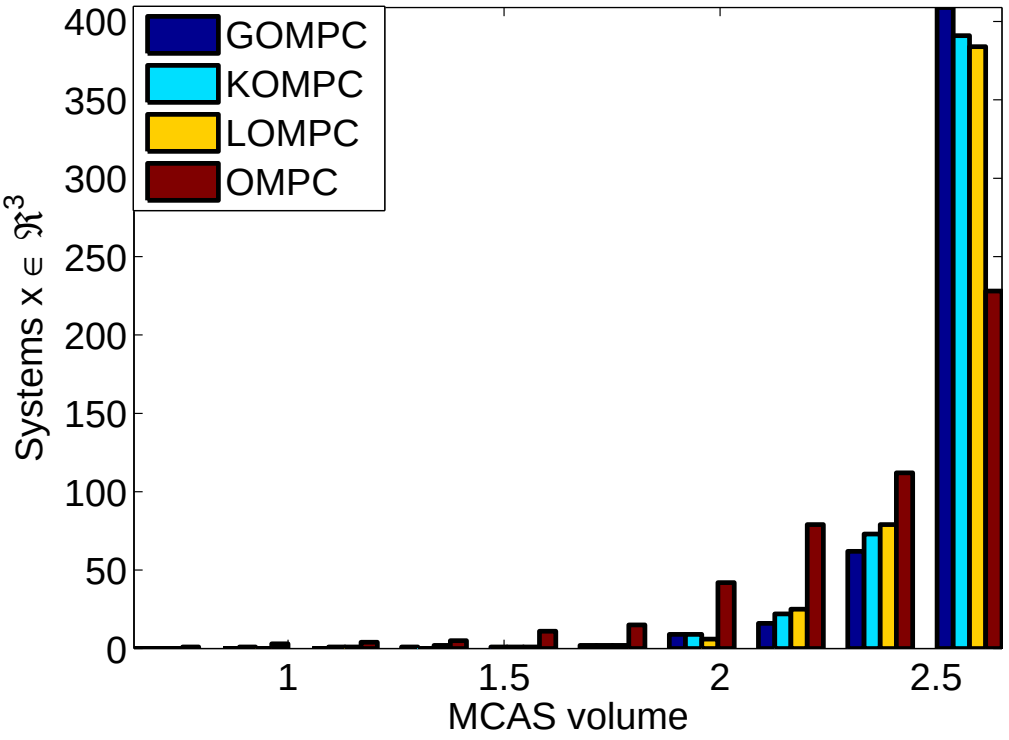

Figure 7.5: Comparison of the MCAS radius for 500 random systems ($x \in \mathbb{R}^3$)

For $3rd$ order random systems using $n_c = 3$, on average GOMPC, KOMPC, LOMPC and OMPC require 38, 43, 47 and 33 inequalities respectively whereas, the maximum number of inequalities are 67, 108, 151 and 45 respectively as shown in Table 7.3. The histogram is shown in Figure 7.7; the number of inequalities of most of the random systems is less than 70, but there are few random system for which the number of inequalities is more than 70 using KOMPC and LOMPC. GOMPC slightly increases the number of inequalities compared to OMPC and enlarges the region of attraction compared to KOMPC, LOMPC and OMPC. From statistical analysis, GOMPC using $3rd$ order dynamics are preferred in general over KOMPC and LOMPC for 3-dimensional systems.

Similarly, for $4th$ order random systems using $n_c = 4$, on average GOMPC, KOMPC, LOMPC and OMPC require 68, 61, 66 and 49 inequalities respectively whereas, the maximum number of inequalities are 139, 157, 182 and 64 respectively as shown in Table 7.4. From a statistical analysis similarly to the $3rd$ order random analysis, GOMPC using $4th$ order dynamics are preferred in general over KOMPC and LOMPC for 4-dimensional systems.

The number of inequalities is compared for 500 random systems using $2nd$, $3rd$ and $4th$

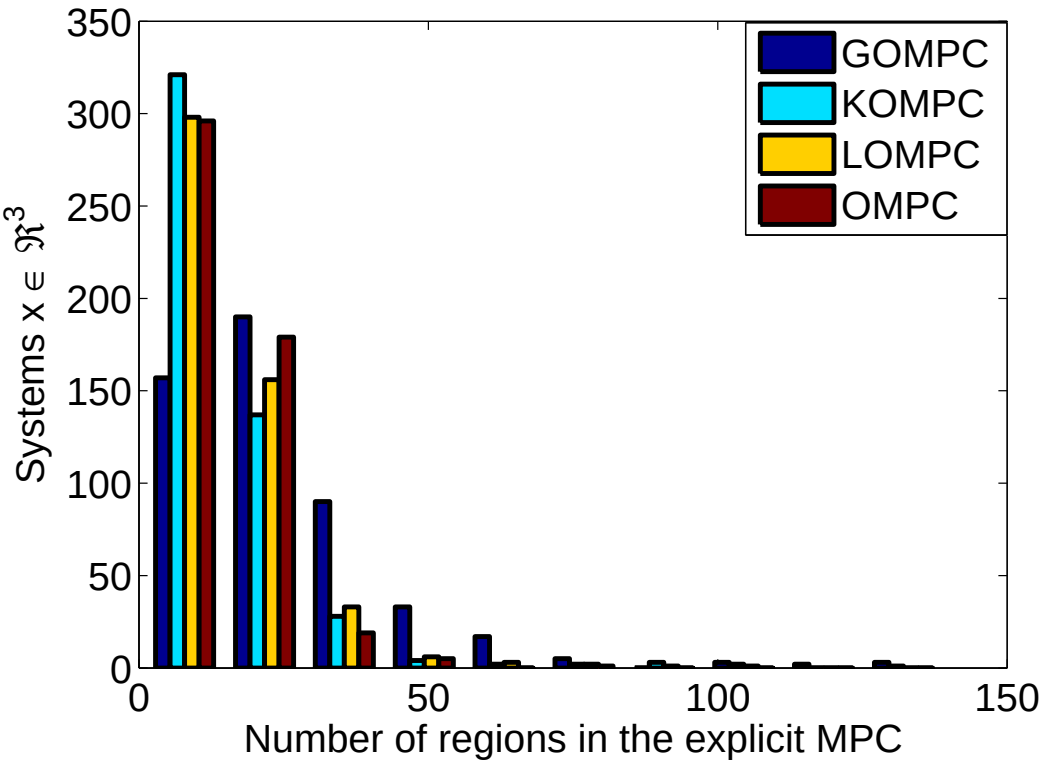

Figure 7.6: Comparison of number of regions for explicit MPC solution for 500 random systems ($x \in \mathbb{R}^3$)

order dynamics. An interesting observation is that the higher order function parameterisation may reduce the number of inequalities using parameterised algorithms. In general parameterised algorithms enlarge the region of attraction at the price of an increase in the number of inequalities compared to OMPC.

7.5.6 Sensitivity of result using different parameter choices

The sensitivity of the result using different parameter choices is presented using 500 single input single output $2nd$ and $3rd$ order dynamic systems. Figure 7.11, 7.12 and 7.13 show the mean MCAS radius, mean inequalities represent MCAS and region in the solution of the explicit MPC respectively using Laguerre parameter variations i.e. $0 \leq p < 1$. All plots start from the OMPC results because for $p = 0$, LOMPC becomes equivalent to OMPC. Here the only Laguerre parameter variation is presented and similar results can be applied to the higher order parameter variations because all parameters vary between 0 and 1.

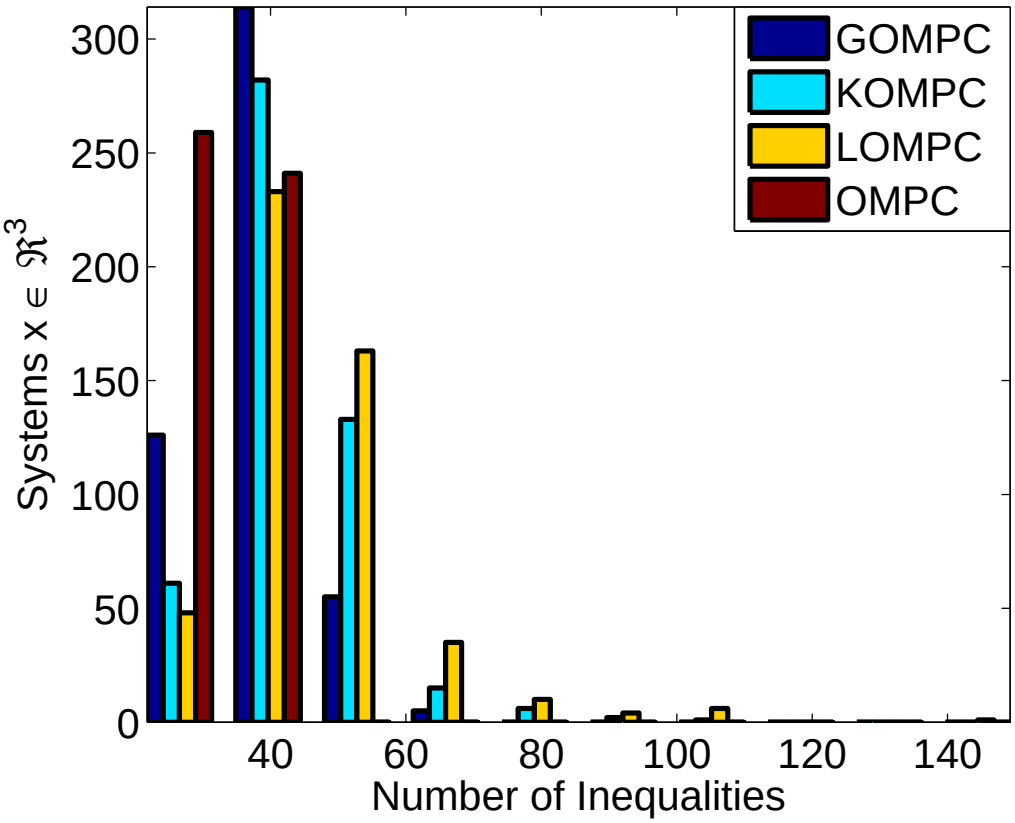

Figure 7.7: Comparison of number of inequalities for 500 random systems ($x \in \mathbb{R}^3$)

The number of inequalities and the number of regions in the solution of explicit MPC increases as the value of p moves away from the origin (i.e. 0 to 1). For both $2nd$ and $3rd$ order dynamic random systems, the inequalities increase with higher rate when $p \geq 0.8$. For $2nd$ order dynamic random systems, the number of regions in the explicit MPC varies from 5 to 7, whereas for $3rd$ order system it increases as p varies from 0 to 1 and varies with higher rates as $p \geq 0.6$. The number of inequalities and the number of regions for $2nd$ order dynamic systems are comparatively less than $3rd$ order dynamic systems.

Figure 7.11 shows the variation of mean MCAS radius for both $2nd$ and $3rd$ order dynamics. The mean MCAS radius increases as p varies up to a certain value of p, after that it starts decreasing. It is interesting to note that for all variations of p, the mean MCAS radius of LOMPC is equal or larger than OMPC.

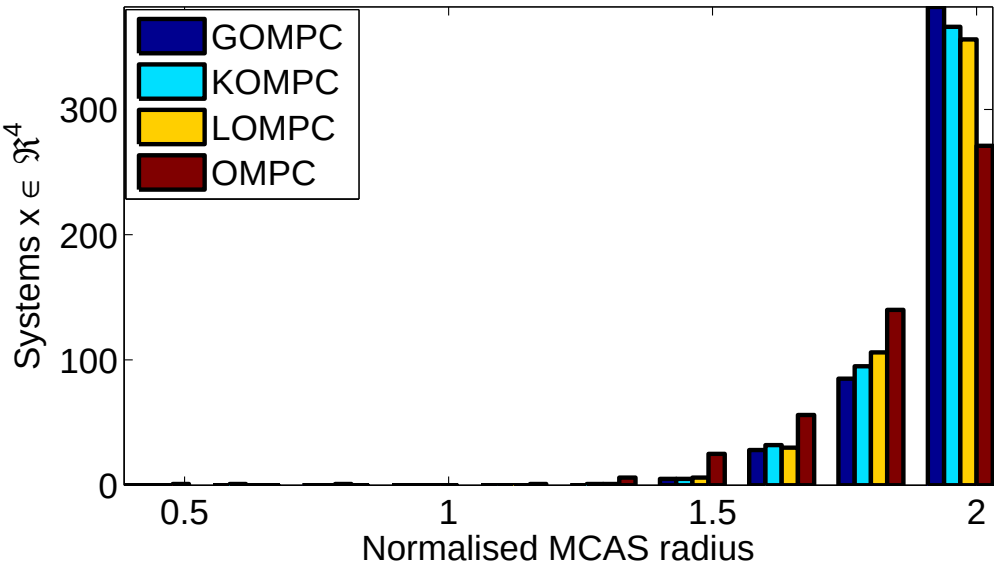

Figure 7.8: Comparison of the MCAS radius for 500 random systems ($x \in \mathbb{R}^4$)

7.5.7 Complexity comparisons using computational time with $n_c = 3$

Table 7.5 lists the computational time required to compute the control law for a fixed number of d.o.f. (i.e. $n_c = 3$) for all randomly generated examples of different sizes using the active set method (ASM) and the generic optimiser solver (quadprog.m from MATLAB). The mean computational time is computed using initial states which are feasible for all algorithms. Table 7.5 shows that alternative parameterisation approaches require an insignificant difference in the number of inequalities (except 4-dimensional example) and computational time using the active set method.

It is well known that the primary factor to affect the computational time is due to the number of constraints using the similar number of d.o.f.. It is already shown in Section 7.5.5 using statistical analysis that in general parameterised algorithms increase the number of inequalities which may compromise the computational time.

7.6 MCAS radius vs computational load

In many cases, the use of generalised functions in predictive control has proven to be a very effective for enlarging the region of attraction, while keeping the number of d.o.f. the same. However, the key question is whether the strategy is computationally better

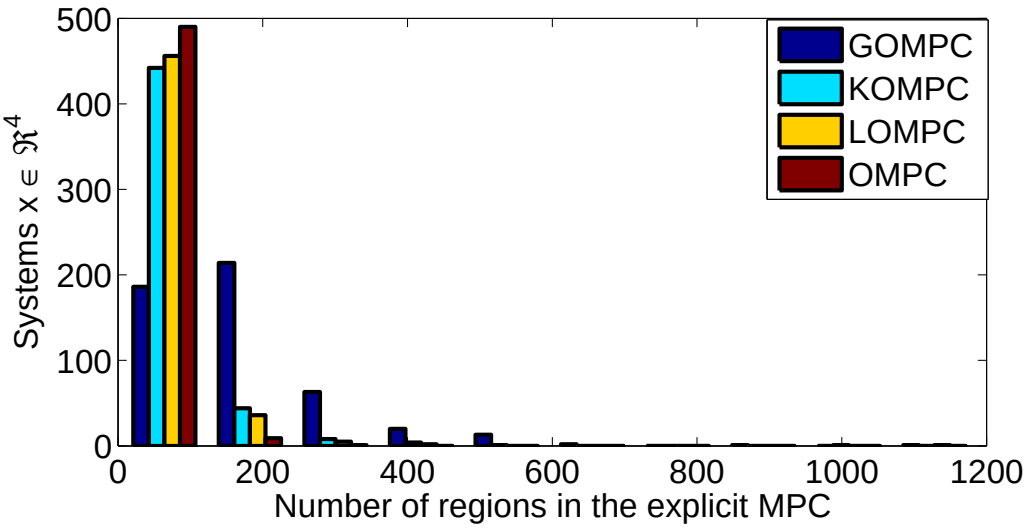

Figure 7.9: Comparison of number of regions for explicit MPC solution for 500 random systems ($x \in \mathbb{R}^4$)

than just increasing the number of d.o.f. available to a standard OMPC algorithm. The QPs in OMPC have a lot of structure and, by exploiting this structure, one is able to compute the control action faster than using a generic optimiser.

In terms of computational load, in general the global region of attraction using GOMPC requires fewer d.o.f. as compared with the OMPC algorithm and this is a useful benefit given any practical limitations on the number of d.o.f. for real time implementation. A $4th$ dimensional example is considered to compare the global region of attraction vs the computational load.

Example $-$ **($x \in \mathbb{R}^4$)**

Consider a 4th dimensional (i.e. $x \in \mathbb{R}^4$) linear system

$$A = \begin{bmatrix} 0.9146 & 0 & 0.0405 & 0.1000 \\ 0.1665 & 0.1353 & 0.0058 & -0.2000 \\ 0 & 0 & 0.1353 & 0.5000 \\ 0 & 0 & 0.1353 & 0.8000 \end{bmatrix}; \; B = \begin{bmatrix} 0.0544 & -0.0757 \\ 0.0053 & 0.1477 \\ 0.8647 & 0 \\ 0.5000 & 0.2000 \end{bmatrix};$$

$$C = \begin{bmatrix} 1.7993 & 13.2160 & 0 & 0.1000 \\ 0.8233 & 0 & 0 & -0.3000 \end{bmatrix}; \; Q = C^T C; \; R = I_{2\times 2}. \tag{7.32}$$

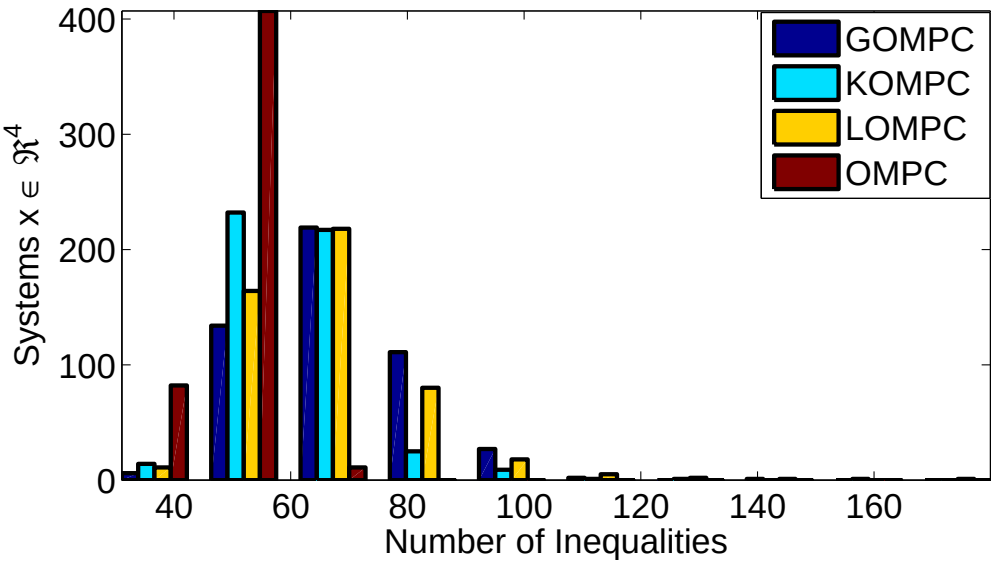

Figure 7.10: Comparison of number of inequalities for 500 random systems ($x \in \mathbb{R}^4$)

The system is subject to input and state constraints

$$|u_k| \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \ |\Delta u_k| \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}; \ |x_k| \leq \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix}. \tag{7.33}$$

The parameterisation dynamics i.e. LOMPC with $p = 0.55$, KOMPC with $(a, b) = (0.55, 0.5)$, and GOMPC ($3rd$ order) with $(a, b, c) = (0.6, 0.55, 0.5)$ are selected in the vicinity of the underlying closed loop stable system poles.

The results of the $4th$ dimensional system are shown in Figure 7.14 and Table 7.6. Figure 7.14 shows the normalised MCAS radius increases as varying d.o.f. for all algorithms. It is clear that GOMPC has a larger MCAS than OMPC, LOMPC and KOMPC for the same number of d.o.f.. Moreover, GOMPC gets to within 100% of the global MCAS with just 3 d.o.f. whereas, KOMPC requires about 4 d.o.f., LOMPC requires 5 d.o.f. and OMPC requires 10 d.o.f.. Clearly alternative parameterisation algorithms required fewer d.o.f. to have global region of attraction.

The computational load is compared with the global region of attraction in Table 7.6. The computational load is compared in terms of number of d.o.f., number of regions, a number of constraints and computational time using both active set method and quadprog.m to achieve the global regions of attraction. The GOMPC has fewer d.o.f. and slightly more

Table 7.2: Statistical Analysis of 2nd dimensional Example for $n_c = 2$

MCAS radius					
Algorithm	minimum	maximum	mean	median	std
OMPC	1.493	24	18.96	22.07	6.015
LOMPC	2.598	24	21.14	24	4.752
KOMPC	3.032	24	21.7	24	4.203
Number of regions in the explicit MPC					
Algorithm	minimum	maximum	mean	median	std
OMPC	1	21	5.216	5	2.959
LOMPC	1	29	5	5	3.203
KOMPC	1	53	5.624	5	4.282
Number of inequalities to describe MCAS					
Algorithm	minimum	maximum	mean	median	std
OMPC	10	25	16.47	17	2.62
LOMPC	10	263	23.32	19	19.16
KOMPC	10	34	15.87	15	3.776

constraints as compared with KOMPC, LOMPC and OMPC.

7.6.1 Implicit implementation

In terms of implicit implementation, alternative function parameterisation including Laguerre, Kautz and 3*rd* order function requires few d.o.f. to represent the global region of attraction. GOMPC requires just 3 d.o.f. to achieve global region of attraction while utilising a computational inexpensive optimisation.

7.6.2 Explicit Implementation

The number of regions required to represent the global region of attraction using alternative (i.e. Laguerre, Kautz and 3rd order) function parameterisation is less as compared with OMPC. GOMPC reduce the number of regions and therefore reduce the online computational burden (as this is correlated to the number of online set membership tests).

Table 7.3: Statistical Analysis of 3rd dimensional Example for $n_c = 3$

MCAS radius					
Algorithm	minimum	maximum	mean	median	std
OMPC	0.6151	2.681	2.344	2.4341	0.3431
LOMPC	1.172	2.681	2.551	2.633	0.195
KOMPC	1.009	2.681	2.552	2.636	0.1976
GOMPC	1.252	2.681	2.575	2.659	0.1771
Number of regions in the explicit MPC					
Algorithm	minimum	maximum	mean	median	std
OMPC	1	73	14.08	13	8.085
LOMPC	1	99	14.98	13	11.29
KOMPC	1	131	14.82	11	13.57
GOMPC	1	139	24.47	21	18.84
Number of inequalities to describe MCAS					
Algorithm	minimum	maximum	mean	median	std
OMPC	23	45	33.31	33	3.498
LOMPC	23	151	47.01	45	13.46
KOMPC	24	108	43.34	43	10.02
GOMPC	20	67	38.38	38	7.118

7.6.3 Summary

The computational load is compared with the global MCAS radius in Table 7.6. An interesting observation is that the primary factor to affect the computational load is due to the number of constraints. It appears from Table 7.6 that alternative (i.e. Laguerre, Kautz and generalised function) parameterisation approaches require insignificant difference in computational time and reduce the number of regions but with a few d.o.f. as compared with OMPC. However generalised parameterisation may have a tall-skinny matrix (as shown in Table 7.6 with global MCAS using few d.o.f.) which improves the storage requirement.

7.7 Conclusion

This chapter has shown the potential benefits of generalised functions as an alternative parameterisation for improving the computational burden in optimal MPC algorithms with a fixed number of d.o.f.. The computational analysis is done for both implicit and explicit implementations. Extensive simulation examples clearly re-affirm the message

Table 7.4: Statistical Analysis of 4th dimensional Example for $n_c = 4$

MCAS radius					
Algorithm	minimum	maximum	mean	median	std
OMPC	0.3623	2.054	1.859	1.909	0.1834
LOMPC	0.8066	2.054	1.928	1.977	0.1367
KOMPC	0.5378	2.054	1.933	1.978	0.1407
GOMPC	1.396	2.054	1.95	1.999	0.1236
Number of regions in the explicit MPC					
Algorithm	minimum	maximum	mean	median	std
OMPC	5	247	43.2	35	28.46
LOMPC	5	1075	60.52	43	71.05
KOMPC	7	1042	75.27	55	75.76
GOMPC	11	1187	176	145	122.5
Number of inequalities to describe MCAS					
Algorithm	minimum	maximum	mean	median	std
OMPC	36	64	49.04	49	4.805
LOMPC	37	182	65.94	63.5	14.65
KOMPC	29	157	60.94	60	11.66
GOMPC	31	139	67.99	67	13.4

that for a fixed and low number of d.o.f., GOMPC, KOMPC and LOMPC enlarge the region of attraction than OMPC. However, in the case of the same number of d.o.f. for both explicit and implicit implementations, then one finds that OMPC may still be competitive in terms of computational load but may have a small region of attraction. In contrast, for the case of the global region of attraction, and using as many degrees of freedom as are required, then GOMPC, KOMPC and LOMPC are computationally efficient. For a number of case studies, all alternative algorithms may enlarge the region of attraction using few numbers of d.o.f., number of regions and slightly larger computational time as compared with OMPC.

Table 7.5: Computational time for $n_c = 3$

4-Dimensional Example			
Algo.	Ineq.	ASM (ms)	Quadprog (ms)
OMPC	64	28	252
LOMPC	150	31	255
KOMPC	149	31	255
GOMPC	127	31	253
10-Dimensional Example			
Algo.	Ineq.	ASM (ms)	Quadprog (ms)
OMPC	118	39	254
LOMPC	126	40	257
KOMPC	126	40	259
GOMPC	144	41	267
16-Dimensional Example			
Algo.	Ineq.	ASM (ms)	Quadprog (ms)
OMPC	198	42	292
LOMPC	192	41	303
KOMPC	191	39	305
GOMPC	202	40	327
30-Dimensional Example			
Algo.	Ineq.	ASM (ms)	Quadprog (ms)
OMPC	206	39	257
LOMPC	206	32	213
KOMPC	206	34	210
GOMPC	206	34	213

Table 7.6: Computational Complexity Vs Global MCAS radius

4-Dimensional Example					
Algo.	Ineq.	ASM (ms)	quadprog.m (ms)	Number of regions	n_c
OMPC	150	67.6	419	301	10
LOMPC	171	74.9	429	265	5
KOMPC	164	66.5	416	149	4
GOMPC	168	74.1	424	105	3

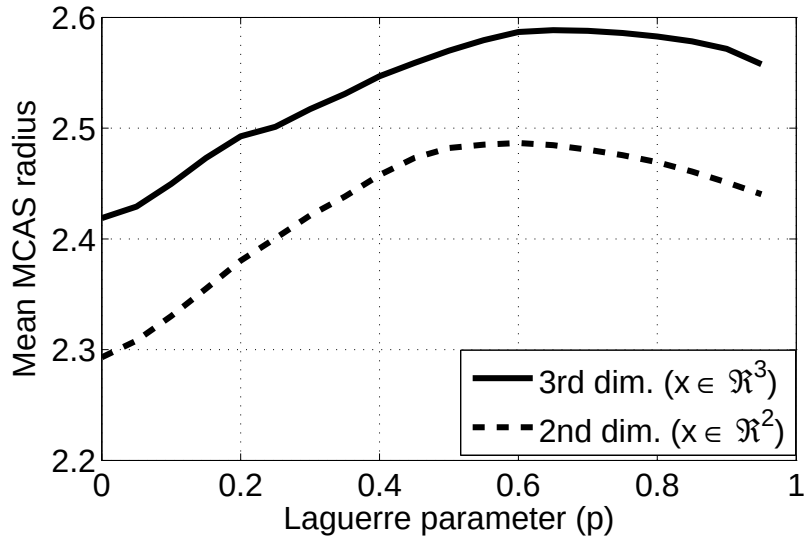

Figure 7.11: Comparison of mean MCAS radius for 500 random systems (for $x \in \mathbb{R}^2$ and $x \in \mathbb{R}^3$) using Laguerre parameter variations.

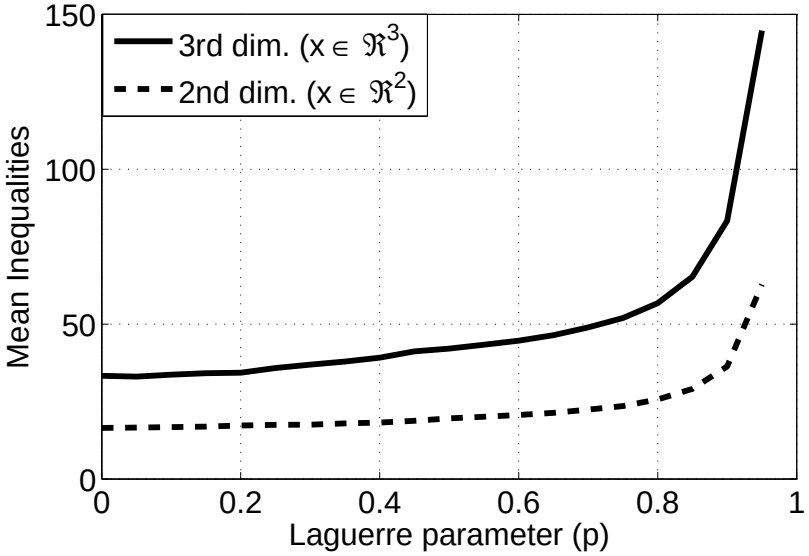

Figure 7.12: Comparison of mean inequalities to represent MCAS for 500 random systems (for $x \in \mathbb{R}^2$ and $x \in \mathbb{R}^3$) using Laguerre parameter variations.

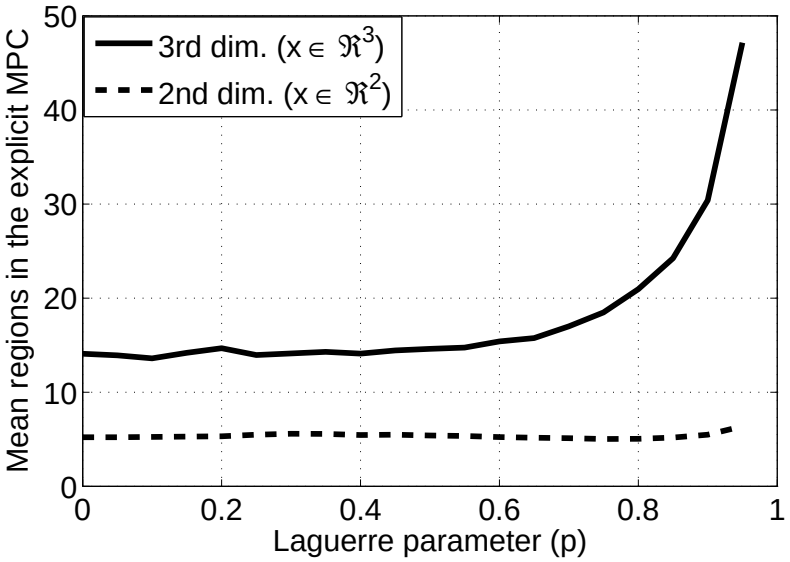

Figure 7.13: Comparison of mean regions in the explicit MPC for 500 random systems (for $x \in \mathbb{R}^2$ and $x \in \mathbb{R}^3$) using Laguerre parameter variations.

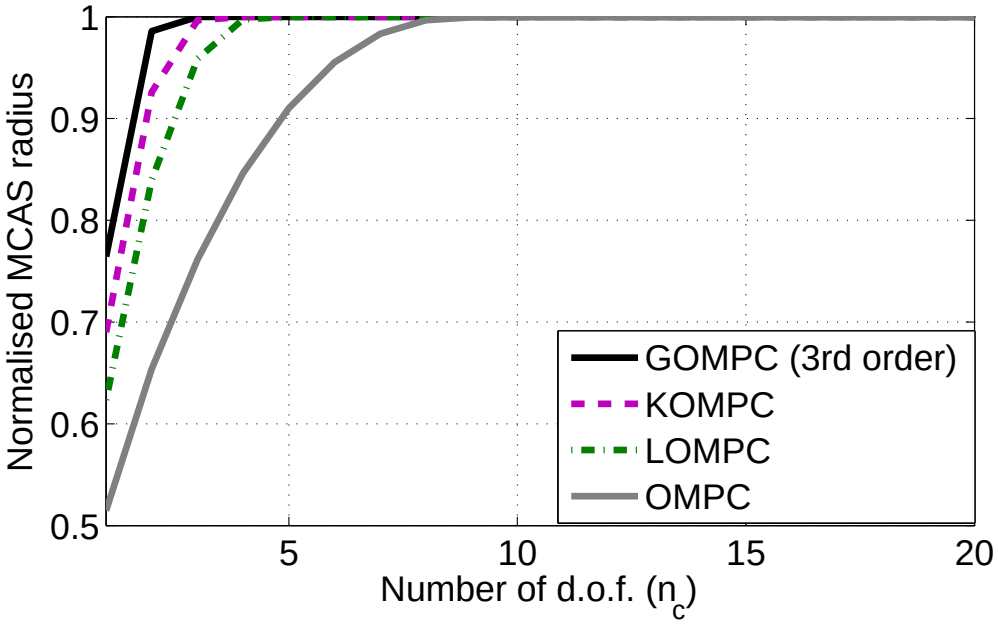

Figure 7.14: Comparison of normalised average MCAS radius as total number of d.o.f. (n_c) vary for all algorithms for 4th dimension system ($x \in \mathbb{R}^4$) in (7.32)

Part III

An Efficient Robust model predictive control using generalised function parameterisation

Chapter 8

Robust predictive control using generalised function parameterisation

This chapter demonstrates the efficacy of more flexible parameterisation of the degrees of freedom within a robust MPC algorithm. Flexible parameterisation has been shown to simplify the trade off within MPC algorithms for the nominal case. This chapter extends that work to the robust scenario and shows that similar benefits accrue and moreover, the increase in complexity of the robust case as compared to the nominal case is much less than might be expected. There are two key contributions: firstly to propose a polyhedral robust control invariant set of an augmented system using generalised function parameterisation; Secondly to propose an algorithm for the robust MPC using the generalised function parameterisation that enables the use of polyhedral robust control invariant set to enlarge the region of attraction. It is also shown that the proposed algorithm has standard convergence and feasibility guarantees. The improvements, with respect to a conventional algorithm, are demonstrated by numerical examples.

This chapter is organised as follows: Section 8.2 gives the necessary background about nominal MPC, generalised function parameterisation for an optimal MPC and robust MPC. Section 8.3 discusses alternative parameterisations within Robust MPC using a generalised function. An algorithm is proposed for Robust MPC using the generalised function parameterisation. Section 8.4 discusses two observations which may help to design an efficient generalised based robust MPC algorithm. Comparisons between the

existing Robust MPC (RMPC) and the new proposed algorithms are given in Section 8.5 using numerical examples. Finally Section 8.6 gives the conclusion of the chapter.

8.1 Introduction

There is substantial interest within the MPC research community how to develop algorithms to deal with nonlinearity or uncertainty, in particular because formal consideration of these issues can lead to substantial computation and/or complexity. The main aim of this chapter is to contribute to research which enlarge the region of attraction while tackling the robust case, perhaps at some small loss of optimality. Specifically, the focus is on the potential of more flexible parameterisation of the degrees of freedom (e.g. Section 5.3) to enable enlargement of the regions of attraction in the uncertain case without too much detriment to performance, optimality and the computational burden.

In the nominal case, Laguerre [26], Kautz and generalised parameterisations are able to achieve large regions of attraction while maintaining insignificant performance drop and a relatively low computational burden. This chapter extends the earlier studies in Section 4.4 and 5.3 to the case of parameter uncertainty by using the algorithm of [115] for constructing polyhedral robust positive invariant sets; this enables the online robust MPC algorithm to be based on a standard quadratic program while adding the benefits of improved feasibility due to the change in the parameterisation.

8.2 Background

This section will introduce the background information on dual mode robust MPC and assumptions used in this chapter.

8.2.1 Polyhedral invariant sets for LPV Systems

One can guarantee robust feasibility/stability if one can determine suitable robust control invariant sets [1](or robust feasible invariant polyhedral sets). The choice of invariant set type (polyhedral vs. ellipsoidal) is largely determined by the computational complexity and the resulting region of attraction. Polyhedral invariant sets lead to regions

[1]This chapter only considers the polyhedral positive invariant sets

of attraction that are guaranteed to be larger than those obtained with their ellipsoidal counterparts and lead to an on-line optimisation class (QP) that can be solved efficiently.

Before discussing the robust dual mode MPC, the following two definitions summarise the concept of robust control set invariance:

Definition 8.1 (Robust Positive Invariance Sets)**.** *[185,195] Given a LPV system* (3.46)*, $\mathcal{X}_r \in \mathbb{R}^{n_x}$ is a robust positively invariant if and only if*

$$x_k \in \mathcal{X}_r \quad \Longrightarrow \; x_{k+1} \in \mathcal{X}_r, \;\; \forall [A_k, B_k] \in Co\{[A_1, B_1], ..., [A_m, B_m]\}. \tag{8.1}$$

Definition 8.2 (Robust Control Invariant Set)**.** *[195] The set $\mathcal{X}_r$ is a robust control invariant set of the LPV system* (3.46) *if and only if there exists a feedback u_k such that $\mathcal{X}_r$ is a robust positive invariant set for the closed loop system and $u_k \in \mathbb{U}$, $\forall x_k \in \mathcal{X}_r$, $[A_k, B_k] \in Co\{[A_{1,k}, B_{1,k}], ..., [A_{m,k}, B_{m,k}]\}$.*

In other words, a set $\mathcal{X}_r$ is a robust control invariant if and only if

$$\begin{aligned} x_k \in \mathcal{X}_r \implies \exists \; u_k \in \mathbb{U} \mid x_{k+1} \in \mathcal{X}_r, \\ \forall [A_k, B_k] \in Co\{[A_{1,k}, B_{1,k}], ..., [A_{m,k}, B_{m,k}]\}. \end{aligned} \tag{8.2}$$

This chapter proposes to make use of a maximal feasible robust control invariant polyhedral set for LPV system that can found for (3.46) subject to (3.49) using an algorithm described in [115].

8.2.2 Robust MPC (RMPC): Dual mode MPC for LPV case

Suppose the infinite horizon linear quadratic performance index is given as

$$J_k = \sum_{i=0}^{\infty} \left[x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k} \right], \tag{8.3}$$

with $Q = Q^T \geq 0$ and $R = R^T > 0$ are state and input cost weighting matrices. Then the worst case performance index to be minimised is

$$\tilde{J}_k = \max_{[A_{k+i}, B_{k+i}] \in Co\{\left[A_{1,k+i}, B_{1,k+i}\right], ..., \left[A_{m,k+i}, B_{m,k+i}\right]\}_{i=0,1...}} J_k \tag{8.4}$$

subject to system dynamics for prediction

$$x_{k+i+1|k} = A_{k+i|k}x_{k+i|k} + B_{k+i|k}u_{k+i|k}. \tag{8.5}$$

In order to guarantee closed loop stability, constraints (3.49) must be satisfied along predicted trajectories for all possible future model uncertainty. Mathematically, this requirement can be written as [25]

$$\begin{aligned} &L_x x_{k+i|k} + L_u u_{k+i|k} \leq l, \\ &\forall \left[A_{k+j}, B_{k+j}\right] \in Co\{\left[A_1, B_1\right], \ldots, \left[A_m, B_m\right]\}, \\ &j = 0, 1 \ldots, i-1. \end{aligned} \tag{8.6}$$

The system will be pre-stabilised with a state feedback controller K as was done in [21, 24, 25].

The nominal performance index corresponding to system dynamic at the centre of the $Co \in \{\left[A_1, B_1\right], \ldots, \left[A_m, B_m\right]\}$ is defined as [25]

$$J_{0,k} = \sum_{i=0}^{\infty} \left[x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k} \right] \tag{8.7}$$

where $x_{0,k+i|k}$ is the predicted trajectory of the central system dynamic for $i \geq 0$.

Consider the autonomous state space model [25]

$$\begin{aligned} &z_{k+i+1|k} = \psi_{k+i|k} z_{k+i|k}, \quad z_{k|k} = \begin{bmatrix} x_k \\ \underrightarrow{c}_k \end{bmatrix}, \\ &\psi_{k+i|k} \in Co\{\psi_j,\ j = 1 \ldots, m\}, \quad \psi_j = \begin{bmatrix} \Phi_j & B_j D \\ 0 & G_c \end{bmatrix}, \\ &\Phi_j = A_j - B_j K, \quad G_c \in Co\{G_{c,j},\ j = 1, \ldots, m\}, \\ &x_{k+i|k} = \left[I \ \ 0\right] z_{k+i|k}, \quad u_{k+i|k} = \left[K \ \ D\right] z_{k+i|k} \end{aligned} \tag{8.8}$$

where $z_k \in \mathbb{R}^{n_x + n_u n_c}$, $\underrightarrow{c}_k^T = [c_k^T, c_{k+1}^T, \ldots, c_{k+n_c-1}^T]$, D and G_c are variables that are used to optimise the size of the associated region of attraction. In [21], $D = E$ and $G_c = I_L$ are given in (3.52) and known as Efficient Robust Predictive Control (ERPC). This approach is improved in [24] by varying parameters in the dynamic feedback law and known as generalised ERPC (GERPC). However the formulation in [24] is non-convex and hence there is no guarantee of convergence to solution. In [25] the dynamic feedback law is further optimised and formulated into a convex problem.

8.3 Using generalised function parameterisation within RMPC

A convex formulation is derived in [25] for optimising dynamic feedback laws for constrained linear systems with polytopic uncertainty and need $n_c = n_x$. The generalised prediction dynamics in (8.8) are based on maximal invariant ellipsoidal set which may be conservative in volume for nonsymmetric constraints as shown in Section 5.4. This section further studies the trade off between performance, region of attraction and computational burden using generalised function parameterisation.

The alternative parameterisations proposed in [18, 26], Section 5.3 for the nominal case showed an enlargement in the region of attraction, for the same number of d.o.f.. Therefore, this section seeks to extend the use of the generalised parameterisations to the robust case and thus explore whether similar feasibility benefits are possible or likely. This section will show how such parameterisation can be used to form robust invariant sets and thus deployed in the appropriate robust MPC algorithm. Examples in the next section are used to demonstrate the impact on the maximal stabilisable set.

There are two main ingredients that are necessary to formulate an efficient robust algorithm using generalised function parameterisation. These ingredients are:

1. The first ingredient is the definition of a robust control invariant set based on generalised function parameterisation.
2. The second ingredient is the definition of the predicted cost.

8.3.1 Generalised function based polyhedral robust control invariant set

As in the nominal case, it was noted earlier (e.g. (5.5)) that using the generalised function, one can define the input predictions as $c_{k+i} = G_i^T \underset{\rightarrow}{\rho}_k$ where $G_i = A_G G_{i-1}$. Unpacking this into a different format one gets

$$\underset{\rightarrow}{\rho}_{k+1} = A_G^T \underset{\rightarrow}{\rho}_k, \quad \underset{\rightarrow}{c}_k = [G_0^T, \ldots, G_{n_c-1}^T]^T \underset{\rightarrow}{\rho}_k \tag{8.9}$$

$$u_k = -Kx_k + D \underset{\rightarrow}{c}_k. \tag{8.10}$$

It is clear therefore that this is equivalent to the autonomous system (8.8) where the choice of $D = G_0^T$. From (8.8), the autonomous formulation using generalised function

parameterisation is defined as

$$
\begin{aligned}
& z_{k+1+i|k} = \psi_{k+i|k} z_{k+i|k},\ z_{k|k} = \begin{bmatrix} x_k \\ \underrightarrow{\rho}_k \end{bmatrix}, \\
& \psi_{k+i|k} \in Co\{\psi_j,\ j = 1 \ldots, m\}, \\
& [A_G, G_0] \in Co\{[A_{G,j}, G_{0,j}],\ j = 1, \ldots, m\}, \\
& \psi_j = \begin{bmatrix} \Phi_j & B_j G_0^T \\ 0 & A_G^T \end{bmatrix}.
\end{aligned} \tag{8.11}
$$

These dynamics should fullfill the constraints (3.49) $\forall k$,

$$
\begin{bmatrix} L_x - L_u K & L_u G_0^T \end{bmatrix} z_{k|k} \leq l. \tag{8.12}
$$

Robust constraint handling is represented by an MCAS or $\mathcal{X}_{r_g}$ which is calculated offline with the methodology of [83, 115], but deploying alternative functions, that is equations (8.11, 8.12) within the update model. The key idea used is not dissimilar to the one-step sets popularised in [195], that is to use forwards prediction rather than backwards predictions. This simple change eliminates the combinatorial explosion in the possible number of prediction terms and hence creates a tractable problem as discussed earlier in Chapter 3. This is illustrated in the algorithm 8.1, for further details the reader is referred to [115].

Algorithm 8.1. *Polyhedral Robust Control Invariant Set*

Given a LPV system (8.11) subject to linear constraints (8.12).

1. *Set the initial values for A_S and b_S to*

$$
A_S := \begin{bmatrix} L_x - L_u K & L_u G_0^T \end{bmatrix}; \quad b_S := l. \tag{8.13}
$$

2. *Initialise the index $i := 1$.*

3. *Repeat until i is not strictly larger than the number of rows in A_S.*

   (a) *Select row i from (8.13) (i.e. $A_{S,i}$ and l_i), check whether adding any of the constraints $A_{S,i}\psi_j z_k \leq l_i,\ j = 1, \ldots, m$ to A_S, b_S would decrease the size of*

$\mathcal{X}_r$, by solving the following linear programming (LP) for $j = 1, \ldots, m$

$$\begin{aligned} c_j = \max_{z_k} \quad & A_{S,i}\psi_j z_k - l_i \\ s.t. \quad & A_S z_k \leq b_S. \end{aligned} \tag{8.14}$$

If $c_j > 0$, then add the constraint to A_S, b_S as follows:

$$A_S := \begin{bmatrix} A_S \\ A_{S,i}\psi_j \end{bmatrix}; \quad b_S := \begin{bmatrix} b_S \\ l_i \end{bmatrix}. \tag{8.15}$$

(b) Increment i.

4. End.

Remark 8.3.1. Algorithm 8.1 will terminate in finite steps and only adds constraints and never removes constraints. The algorithm convergence and invariance of the resulting set $\mathcal{X}_{g_r} = \{z_{k|k} : A_S z_{k|k} \leq b_S\}$ is proved similarly as in [115]. After terminating, it is recommended to remove any redundant constraints.

Remark 8.3.2. MAS or $\mathcal{X}_{r_0}$ is calculated using the above algorithm with $[x, c] = [x, 0]$ or using algorithm defined in [115].

8.3.2 Generalised function parameterisation within RMPC

The worst case prediction cost is defined similarly as in (8.4)

$$\tilde{J}_{G,k} = \max_{[A_{k+i},B_{k+i}] \in Co\{[A_{1,k+i},B_{1,k+i}],\ldots,[A_{m,k+i},B_{m,k+i}]\}_{i=0,1\ldots}} J_k, \tag{8.16}$$

using (3.28) and (5.12) the worst cost can be written as

$$\tilde{J}_{G,k} = \max_{[A_{k+i},B_{k+i}] \in Co\{[A_{1,k+i},B_{1,k+i}],\ldots,[A_{m,k+i},B_{m,k+i}]\}_{i=0,1\ldots,n_c-1}} \underset{\rightarrow}{\rho}{}^T_{k+i|k} S_G \underset{\rightarrow}{\rho}_{k+i|k}, \tag{8.17}$$

where $S_G = \sum_{i=0}^{\infty} A_G^i G_0 S G_0^T {A_G^i}^T$.

For completeness, the MCAS is calculated offline using algorithm 8.1 as

$$\mathcal{X}_{r_g} = \{x_k : \exists \underset{\rightarrow}{\rho}_k \ s.t. \ A_S z_k \leq b_S, \ z_k = [x_k, \underset{\rightarrow}{\rho}_k]^T\}, \tag{8.18}$$

and associated MAS is defined as

$$\mathcal{X}_{0_g} = \{x_k : \exists \underset{\rightarrow}{\rho}_k = 0 \ s.t. \ A_S z_k \leq b_S, \ z_k = [x_k, 0]^T\}. \tag{8.19}$$

After calculating the inequalities for invariant set the following RGMPC algorithm 8.2 can be implemented.

Algorithm 8.2. *Robust generalised MPC (RGMPC)*

Off-line

- *Determine the polyhedral robust control invariant set can be found in* (8.8) *subject to* (3.49) *using the algorithm 8.1 [115], in the form*

$$A_S z_k \leq b_S. \tag{8.20}$$

On-line

1. *At each sampling instant, solve the following optimisation problem:*

$$\min_{\underset{\rightarrow}{\rho}_k} \tilde{J}_{G,k} \ \ s.t. \ \ A_S z_k \leq b_S. \tag{8.21}$$

2. *Reconstitute the first value of the predicted input trajectory* u_k *using* $\underset{\rightarrow}{c}_k = [G_0^T, \ldots, G_{n_c-1}^T]^T \underset{\rightarrow}{\rho}_k$ *and* (3.14).

3. *If the unconstrained control law is satisfying the constraints (i.e.* $x_k \in \mathcal{X}_{0_g}$*), the optimising* $\underset{\rightarrow}{c}_k = [G_0^T, \ldots, G_{n_c-1}^T]^T \underset{\rightarrow}{\rho}_k$ *is zero so the control law is* $u_k = -Kx_k$.

4. *End.*

Theorem 8.1. *The RGMPC algorithm has a guarantee of stability and recursive feasibility.*

Proof. Recursive feasibility: The key to RGMPC algorithm is the requirement that

$$x_k \in \mathcal{X}_{r_g} \implies x_{k+1} \in \mathcal{X}_{r_g} \tag{8.22}$$

and in fact to be more precise, one requires that an augmented state including the tail of $\underset{\rightarrow}{c}_k = [G_0^T, \ldots, G_{n_c-1}^T]^T \underset{\rightarrow}{\rho}_k$ remains feasible, that is:

$$A_S \begin{pmatrix} x_k \\ \underset{\rightarrow}{\rho}_k \end{pmatrix} \leq b_S \implies A_S \begin{pmatrix} x_{k+1} \\ \underset{\rightarrow}{\rho}_{tail|k} \end{pmatrix} \leq b_S \tag{8.23}$$

where $\underset{\rightarrow}{\rho}_{tail|k} = [\rho_{k+1|k}^T, \ldots, \rho_{k+n_c-1|k}^T, 0]^T$. Such a guarantee can be established by defining A_S and b_S as robust invariant set using algorithm 8.1 using [115]. Finally note that for all states inside the MAS i.e. $\underset{\rightarrow}{\rho}_k = [0, \ldots, 0]^T$, the unconstrained optimal control law $u_k = -Kx_k$ will feasible.

Prediction cost: The stability proof follows a well accepted route of showing that the tail of the optimum from sampling instant k is a valid choice at sampling instant $k+1$; as this implies a reduction in $\tilde{J}_{G,k}$. A possible choice at the next sampling instant is $\underset{\rightarrow}{\rho}_{k+i|k+1} = \underset{\rightarrow}{\rho}_{k+i|k}$ and hence $\tilde{J}_{G,k+1} \leq \tilde{J}_{G,k} - \rho_k^T G_0 S G_0^T \rho_k$. Hence one can prove convergence using theorem 5.1 that $\tilde{J}_{G,k}$ is monotonically non-increasing $\forall k$ and if $c_k = G_0^T \underset{\rightarrow}{\rho}_k$ is repeatedly zeros, which implies that the state is already inside the terminal region and $\tilde{J}_{G,k} = 0$. Hence, the Lyapunov stability of the origin follows from the fact that the MAS contains the origins in its interior.

□

8.4 Observations

There are two observations which may be helpful for the design engineers to propose an efficient generalised based robust MPC.

8.4.1 Order selection of the generalised parameterisation dynamic

In generalised parameterisations, the higher order prediction dynamics have more flexibility to improve the region of attraction with a limited number of d.o.f.. So there is a clear choice of selecting the order of the parameterisation dynamics.

From the autonomous formulation using generalised parameterisation in (8.11), to fulfill the algebraic relations the dimension of the A_G must be the same as n_c using the same number of d.o.f.. Moreover, the key observation from the augmented model in (8.11) is that $dim(A_G) = n_c$ is an upper bound on the maximum parameterisation dynamics order. The generalised function parameterisation dynamics remove the limitation on the selection of n_c i.e. $n_c \geq n_x$.

8.4.2 Selection of parameterisation poles using closed loop dynamics

In uncertain cases, the optimal selection of using multiobjective optimisation is computationally demanding due to the computations of the robust control invariant set. In this chapter the parameterisation dynamics are selected in the vicinity of the closed loop stable system (i.e. the closed loop stable central system of the $Co \in [A_1, B_1], \ldots, [A_m, B_m]$) pole(s). This is a pragmatic selection similar to the nominal case, albeit with sub optimal parameter value(s). For further discussion on parameter(s) selection the reader is referred to Chapter 6.

8.5 Numerical Examples

This section will illustrate the efficacy of the alternative parameterisation (i.e. Laguerre, Kautz and generalised functions) within the robust MPC algorithm in comparison with GERPC [25] and ERPC [21] using numerical examples. The aim is to compare three aspects: (i) the size of the regions of attraction; (ii) the number of inequalities required to describe the robust MCAS; (iii) closed-loop performance. For the purposes of visualisation, figures are restricted to second order systems for which it is possible to plot regions of attraction. The comparisons are based on 4 systems with $x \in \mathbb{R}^2, x \in \mathbb{R}^3$ and $x \in \mathbb{R}^4$. The alternative parameterisarions are based on Laguerre, Kautz and generalised functions (using $3rd$ order dynamics).

8.5.1 Example 1

Consider a linear uncertain system representing a double integrator with an uncertainty polytope defined by the following two vertices (used in [114] and [83]):

$$\begin{aligned} A_1 &= \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix},\ B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ A_2 &= \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix},\ B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}, \\ Q &= I,\ R = 1,\ n_c = 2,\ \gamma = 10^{10},\ p = \{0.9, 0.8\},\ (a,b) = \{(0.9, 0.5), (0.8, 0.2)\}. \end{aligned} \tag{8.24}$$

The system is subject to input and state symmetric and non-symmetric constraints

Example 1 (a)

$$-1 \leq u_k \leq 1; \qquad -\begin{bmatrix} 5 \\ 5 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}; \tag{8.25}$$

Example 1 (b)

$$-0.5 \leq u_k \leq 1; \qquad -\begin{bmatrix} 7 \\ 7 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 5 \\ 5 \end{bmatrix}. \tag{8.26}$$

Figure 8.1 and 8.2 shows the regions of attraction for RKMPC, RLMPC, GERPC and ERPC for the same number of d.o.f. i.e. $n_c = 2$. For the symmetric case in Example 1 (a), RKMPC has a larger MCAS volume than RLMPC, GERPC and ERPC. Whereas in Figure 8.1, GERPC has a larger region of attraction than RLMPC and ERPC. Similarly as in the nominal case in Section 5.4, for non-symmetric case in Example 1 (b), RKMPC and RLMPC enlarge the region of attraction significantly compared to GERPC.

8.5.2 Example 2

Consider another 2nd order (i.e. $x \in \mathbb{R}^2$) linear uncertain system representing an uncertainty polytope defined by the following two vertices:

$$\begin{aligned} A_1 &= \begin{bmatrix} 0.6 & -0.4 \\ 1 & 1.4 \end{bmatrix},\ B_1 = \begin{bmatrix} 0.2 \\ 0.05 \end{bmatrix}, \\ A_2 &= \begin{bmatrix} 0.6 & -0.5 \\ 1 & 1.4 \end{bmatrix},\ B_2 = \begin{bmatrix} 0.2 \\ 0.5 \end{bmatrix}, \\ Q &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ R = 0.2,\ n_c = 2,\ \gamma = 10^{10},\ p = 0.8,\ (a, b) = (0.8, 0.78). \end{aligned} \tag{8.27}$$

The system is subject to input and state symmetric and non-symmetric constraints

Example 2 (a)

$$-1 \leq u_k \leq 1; \qquad -\begin{bmatrix} 2 \\ 2 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}; \tag{8.28}$$

Example 2 (b)

$$-1.5 \leq u_k \leq 1; \qquad -\begin{bmatrix} 1 \\ 2 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}. \tag{8.29}$$

Figure 8.3 and 8.4 show the regions of attraction for RKMPC, RLMPC, GERPC and ERPC for the same number of d.o.f. i.e. $n_c = 2$. For both symmetric and non-symmetric cases, RKMPC has a larger MCAS volume than RLMPC, GERPC and ERPC as shown in Table 8.1. Whereas in Figure 8.3 and 8.3, there are some initial points in GERPC MCAS which are infeasible for RLMPC and RKMPC algorithms. Similarly as in the nominal case in Section 5.4, for non-symmetric case in Example 1 (b), RKMPC and RLMPC enlarge the region of attraction significantly compared to GERPC.

8.5.3 Example 3

Consider a 3rd order (i.e. $x \in \mathbb{R}^3$) linear uncertain system represented with an uncertainty polytope defined by the following two vertices:

$$A_1 = \begin{bmatrix} 1.4 & -0.1050 & -0.1080 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; B_1 = \begin{bmatrix} 0.2 \\ 0 \\ 0 \end{bmatrix},$$
$$A_2 = \begin{bmatrix} 1.5 & -0.2050 & -0.1080 \\ 2 & 0 & 0 \\ 0 & 1.5 & 0 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0.3 \\ 0 \\ 0 \end{bmatrix}, \tag{8.30}$$
$$Q = I, \; R = 1, \; n_c = 3, \; \gamma = 10^{10}, \; p = 0.68, \; (a, b) = (0.68, 0.05), \; (a_1, a_2, a_3) = (0.65, 0.52, 0.5).$$

The system is subject to input and state symmetric and non-symmetric constraints

Example 3 (a)

$$-1 \leq u_k \leq 1; \qquad -\begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}; \tag{8.31}$$

Example 3 (b)

$$-0.5 \leq u_k \leq 1; \qquad -\begin{bmatrix} 7 \\ 4 \\ 3 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}. \tag{8.32}$$

Table 8.1 shows MCAS volume for RGMPC ($3rd$ order) RKMPC, RLMPC, GERPC and ERPC for the same number of d.o.f. i.e. $n_c = 3$. Alternative parameterisations (i.e. RGMPC and RKMPC) enlarge the region of attraction compared to GERPC and ERPC.

8.5.4 Example 4

Consider a 4th order (i.e. $x \in \mathbb{R}^4$) linear uncertain system representing an uncertainty polytope defined by the following two vertices:

$$A_1 = \begin{bmatrix} 0.900 & -0.105 & 0.108 & 0.200 \\ 0.600 & 0 & 0 & -0.100 \\ 0 & 0.800 & 0 & 0.300 \\ 0 & 0 & 0.800 & 0 \end{bmatrix}, \; B_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$
$$A_2 = \begin{bmatrix} 0.900 & -0.105 & 0.108 & 0.2 \\ 0.600 & 0 & 0 & -0.150 \\ 0 & 0.900 & 0 & 0.300 \\ 0 & 0 & 0.800 & 0 \end{bmatrix}, \; B_2 = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{8.33}$$
$$Q = I, \; R = 0.5, \; n_c = 4, \; \gamma = 10^{10}, \; p = 0.5, \; (a, b) = (0.5, 0.2),$$
$$(a_1, a_2, a_3) = (0.5, 0.2, 0.1), \; (a_1, a_2, a_3, a_4) = (0.5, 0.45, 0.2, 0.1).$$

The system is subject to input and state symmetric and non-symmetric constraints

Example 4 (a)

$$-1 \leq u_k \leq 1; \qquad -\begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix}; \tag{8.34}$$

Example 4 (b)

$$-0.5 \leq u_k \leq 1; \qquad -\begin{bmatrix} 15 \\ 7 \\ 4 \\ 5 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 10 \\ 10 \\ 10 \\ 10 \end{bmatrix}. \tag{8.35}$$

Table 8.1 shows MCAS volume for RGMPC ($4th$ order), RGMPC ($3rd$ order), RKMPC, RLMPC, GERPC and ERPC for the same number of d.o.f. i.e. $n_c = 4$. For symmetric

Table 8.1: Comparison of robust MCAS volume and number of inequalities

Symmetric Constraints								
Algorithm	Example 1		Example 2		Example 3		Example 4	
	Vol.	Ineq.	Vol.	Ineq.	Vol.	Ineq.	Vol.	Ineq.
ERPC	67.21	22	1.3651	18	140.06	38	1.53×10^5	80
GERPC	94.74	40	3.3058	40	210.78	76	1.54×10^5	42
RLMPC	94.51	34	3.8180	74	201.20	50	1.54×10^5	54
RKMPC	95.00	30	4.667	116	220.08	32	1.54×10^5	32
RGMPC (3rd order)	-	-	-	-	221.74	56	1.54×10^5	45
RGMPC (4th order)	-	-	-	-	-	-	1.54×10^5	43
Non-symmetric Constraints								
Algorithm	Example 1		Example 2		Example 3		Example 4	
	Vol.	Ineq.	Vol.	Ineq.	Vol.	Ineq.	Vol.	Ineq.
ERPC	63.18	21	2.1281	18	56.84	43	3.92×10^4	75
GERPC	102.18	97	3.5304	37	78.03	80	4.12×10^4	90
RLMPC	135.50	57	3.8908	76	99.19	96	5.07×10^4	90
RKMPC	136.59	30	4.4557	109	99.91	58	5.19×10^4	79
RGMPC (3rd order)	-	-	-	-	102.97	53	5.56×10^4	81
RGMPC (4th order)	-	-	-	-	-	-	5.56×10^4	76

constraint, alternative parameterisations and GERPC have the same volume of MCAS and RKMPC is the best choice with fewer inequalities. Similar to the previous examples, for non-symmetric constraints alternative paramterisations enlarge the region of attraction with fewer number of inequalities compared to GERPC.

8.5.5 Regions of attraction

The regions of attraction for Examples 1 and 2 are plotted in figure 8.1, 8.2 and 8.3, 8.4 respectively. It is clear from the figures that the use of alternative (Laguerre, Kautz function) parameterisation techniques within robust MPC algorithms enlarge the region of attraction for non-symmetric constraints. Table 8.1 shows the MCAS volume (volume of projection) comparisons for Examples 1-4 using ERPC, GERPC, RLMPC, RKMPC, RGMPC ($3rd$ order) and RGMPC ($4th$ order) algorithms. Alternative (Kautz and generalised function) parameterisations enlarge the region of attraction significantly (specifically for non-symmetric constraints) and thus, based solely on volume considera-

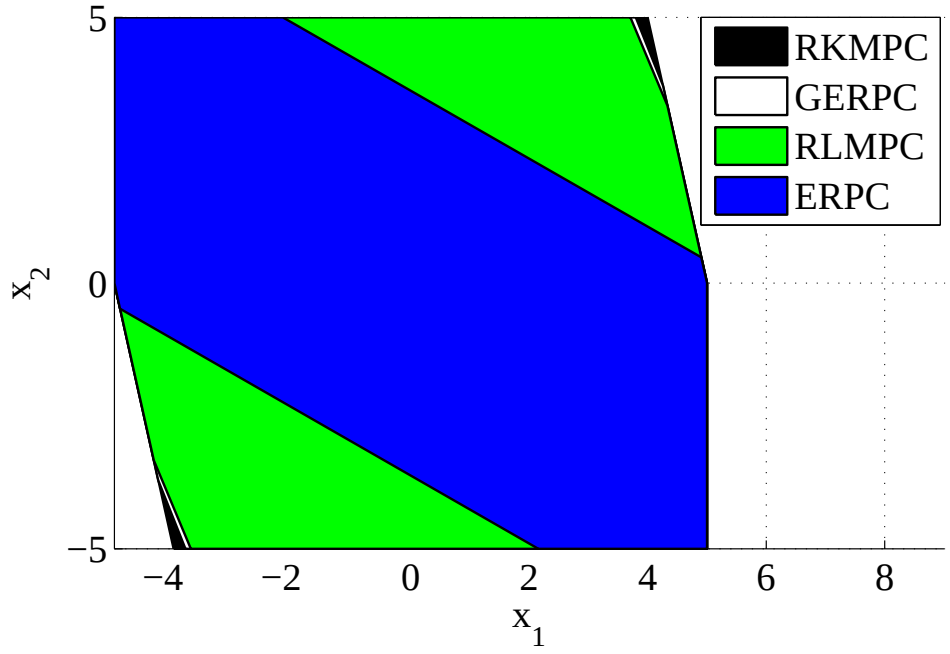

Figure 8.1: Robust MCAS regions for model (8.24) with symmetric constraints

tions and as expected, alternative parameterisation based algorithms are to be preferred in a robust scenario.

From Table 8.1 and Figure 8.2 and 8.4 shows that for non-symmetric constraints, alternative parameterisations significantly enlarge the region of attraction than GERPC.

8.5.6 Closed-loop performance

The closed-loop performance of RGMPC, RKMPC, RLMPC and ERPC algorithms is contrasted for 200 random initial states. To make the comparison meaningful therefore all consider the initial conditions which lie in the ERPC region of attraction. Table 8.2 shows the average of $\tilde{J}_k$ using (8.4) for a set of 200 random initial points. Table 8.2 represents an average cost comparison for both symmetric and non-symmetric constraints.

For example 1, ERPC, RKMPC and RLMPC provide 36.26 and 32.36 cost on average for both symmetric and non-symmetric constraints respectively. In example 2, ERPC, RKMPC and RLMPC provide 5.07, 5.57 and 5.73 cost on average respectively for symmetric constraints. For non-symmetric constraints, ERPC, RKMPC and RLMPC provide 4.70 cost on average.

For example 3, ERPC, RLMPC, RKMPC and RGMPC provide 21.30, 31.17, 31.23 and

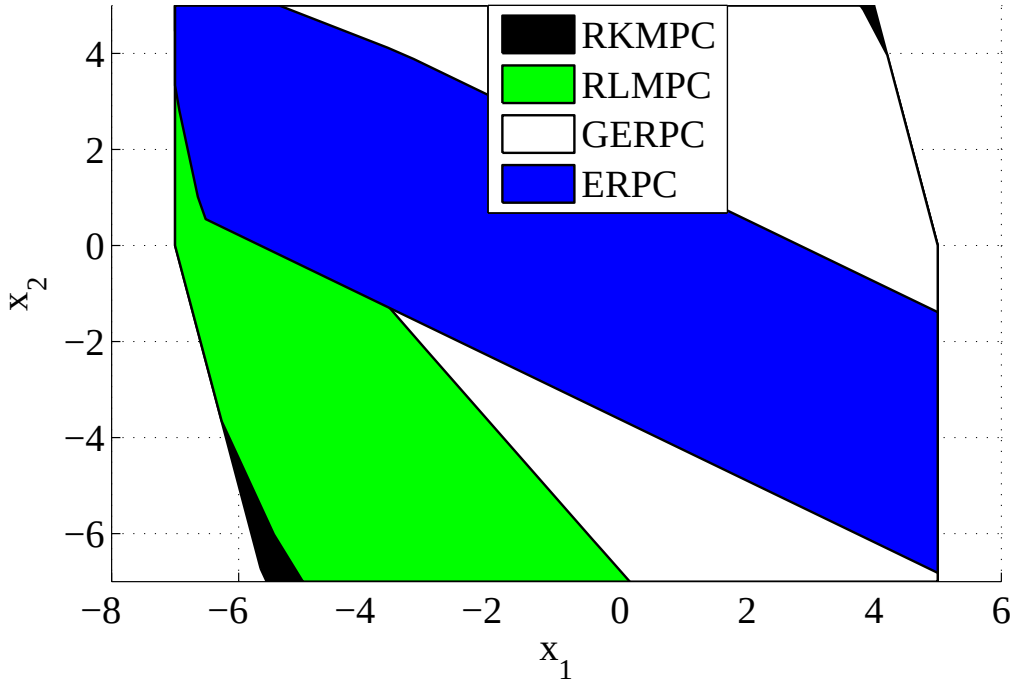

Figure 8.2: Robust MCAS regions for model (8.24) with non-symmetric constraints

31.53 cost on average for symmetric constraints respectively. For non-symmetric constraints, ERPC, RLMPC, RKMPC and RGMPC provide 15.28, 16.77, 16.78 and 16.77 cost on average respectively.

In example 4, ERPC, RLMPC, RKMPC, RGMPC ($3rd$ order) and RGMPC ($4th$ order) provide 42.70, 43.06, 43.06, 43.05 and 45.53 cost on average using symmetric constraints. For non-symmetric constraints, ERPC, RLMPC, RKMPC, RGMPC ($3rd$ order) and RGMPC ($4th$ order) provide 19.52, 19.53, 19.53, 19.67 and 19.74 cost on average respectively.

Table 8.2 shows for both symmetric and non-symmetric constraints, alternative algorithms (i.e. RGMPC, RKMPC and RLMPC) enlarge the region of attraction without too much degradation to the closed loop performance as compared to ERPC.

8.5.7 Computational Complexity

For completeness, it is important to compare the number of inequalities required to describe the robust MCAS as the complexity of these set descriptions has an impact on the online computational burden, the more inequalities the higher the computational

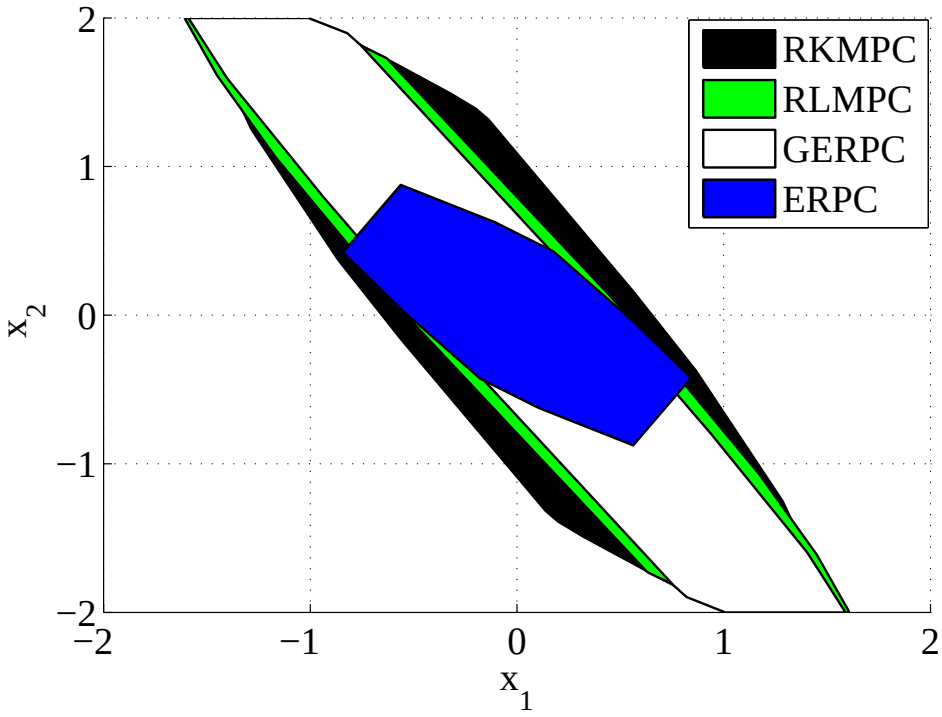

Figure 8.3: Robust MCAS regions for model (8.27) with symmetric constraints

burden in solving the associated QP optimisation (this chapter does not discuss issues linked to the exploitation of structure and efficient QP optimisers). The number of inequalities to define $\mathcal{X}_r$ is compared with the same number of d.o.f. in Table 8.1. The number of inequalities with parameterisation based algorithms are more in comparison with ERPC algorithm. However, it is notable that for non-symmetric constraints, the number of inequalities for GERPC algorithm is a slightly larger in comparison with RGMPC ($3rd$ order), RKMPC and RLMPC algorithms.

It is clear from Table 8.1 that alternative function parameterisations (i.e. RGMPC, RKMPC and RLMPC) enlarge the region of attraction without too much compromise the computational benefits in terms of number of inequalities.

8.5.8 Summary

The results shown in Figure 8.1, 8.2, 8.2, 8.4 and in Table 8.1 make it clear that RGMPC, RKMPC and RLMPC provide an alternative proposal to [25] to further enlarge the region of attraction. In [25], optimisation dynamics are calculated using maximal ellipsoidal invariant sets and can therefore only cope with asymmetric constraints in a conservative

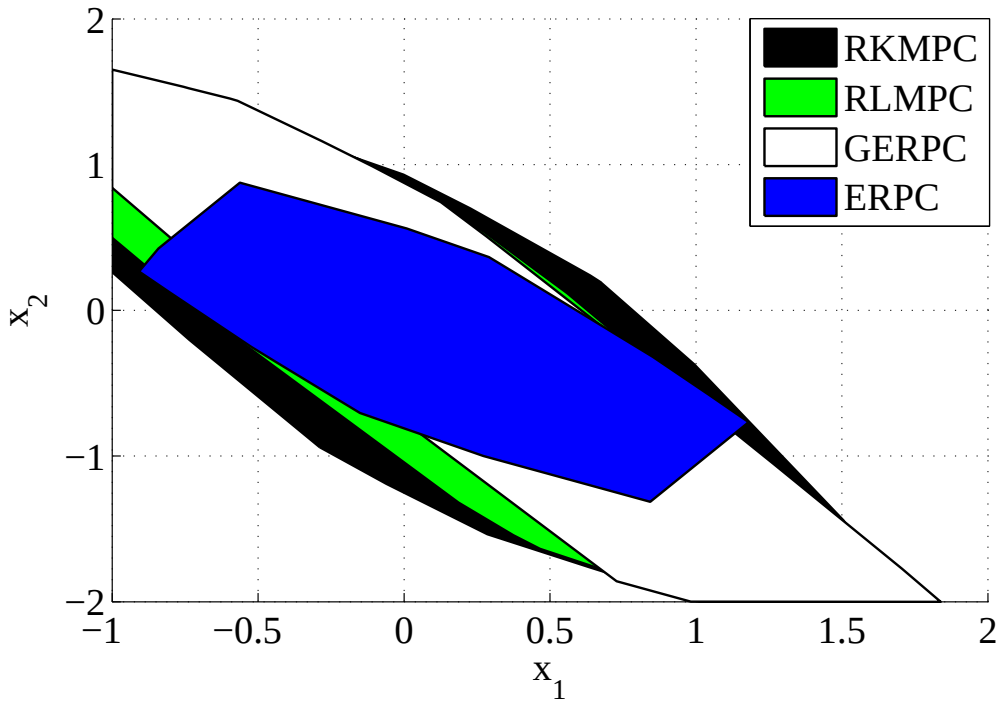

Figure 8.4: Robust MCAS regions for model (8.27) with non-symmetric constraints

way. Figure 8.2, 8.4 and Table 8.1 show that the resulting region of attraction is also conservative for non-symmetric constraints. From Table 8.2, it concluded that alternative parameterisatons enlarge the region of attraction without too much degradation to the closed-loop performance.

These results are based on pragmatic choices for the parameters in RGMPC, RKMPC and RLMPC. Further improvements both in the region of attraction and number of inequalities are possible by tailoring these parameters in the context.

8.6 Conclusion

The main contribution of this chapter was to extend robust MPC algorithms to make use of alternative parameterisations of the d.o.f. and to consider the impact of doing so. Different alternative parameterisation functions including Laguerre, Kautz and higher order functions are embedded within the robust MPC approach; the main requirement for this is to show how a robust control invariant set can be computed with different parameterisations of the d.o.f.. The examples demonstrate that, for a fixed number of d.o.f., in many cases much parameterisation may enlarge the region of attraction without

Table 8.2: Comparison of average runtime cost

Symmetric Constraints				
Algorithm	Example 1	Example 2	Example 3	Example 4
ERPC	36.2647	5.0694	21.2959	42.6982
RLMPC	36.2647	5.5739	31.1691	43.0642
RKMPC	36.2647	5.7324	31.2278	43.0642
RGMPC (3rd order)	-	-	31.5264	43.0499
RGMPC (4th order)	-	-	-	43.5258
Non-symmetric Constraints				
Algorithm	Example 1	Example 2	Example 3	Example 4
ERPC	32.3604	4.6973	15.2774	19.5238
RLMPC	32.3604	4.6973	16.7714	19.5336
RKMPC	32.3604	4.6973	16.7800	19.5338
RGMPC (3rd order)	-	-	16.7664	19.6738
RGMPC (4th order)	-	-	-	19.738

any significant change to the number of inequalities required to describe the robust control invariant set than GERPC.

Chapter 9

Robust triple mode predictive control using flexible function parameterisation

This chapter presents an **original contribution** to the thesis. It reviews triple mode Predictive control of linear time invariant and uncertain systems, considers the analogies with new approaches to conventional dual mode MPC algorithms deploying more flexible function parameterisation. It is shown that there are strong analogies and moreover, that using the Laguerre, Kautz and generalised functions insights within a triple mode MPC approach may significantly enlarge the region of attraction. There are many cases where such an approach is an improvement on earlier work and thus this avenue of research is worth pursuing for both nominal and robust scenarios. The improvements, with respect to an existing algorithm, are demonstrated by numerical examples.

This chapter is organised as follows: Section 9.1 presents the introduction and motivation of the chapter; Section 9.2 presents the background and brief overview of the triple mode MPC algorithms; Section 9.3 presents novel triple mode MPC algorithms using the generalised function dynamics; Section 9.4 presents a novel robust triple mode MPC using generalised functions parameterisation for both implicit and explicit selection of middle mode; Section 9.5 presents numerical examples; and finally Section 9.6 gives the conclusion of the chapter.

9.1 Introduction

In the early days of MPC it was noted that MPC included some inherent robustness and indeed this is sufficient for many of the widespread industrial applications, albeit the applications come without formal guarantees or a more systematic robust design [77,78]. Later authors considered how to include robustness requirements explicitly; generally these require quantification of the uncertainty into formal bounds, for example on parameter variation and the magnitude of disturbance signals. Given these bounds, min-max types of optimisation design are feasible and many algorithms have been proposed; this chapter will develop the type of robust MPC approach used in [83, 115] which uses a linear parameter varying (LPV) model to represent parameter uncertainty.

One suggestion that is still relatively underexplored in the literature is the concept of triple mode control [112–114]. In this strategy one recognises that large regions of attraction in conjunction with good performance often imply nonlinear or linear time varying (LTV) prediction dynamics [177]. In fact it is known that the optimal law is piecewise affine, but that introduces a directional dependence which is a further complication this chapter wishes to avoid to achieve simplicity. Hence, a sensible objective is to find a suitable and fixed LTV control law which enlarges the region of attraction without too much detriment to performance.

The first triple mode controller [112, 113] used the algorithm of [21] to specify the additional mode of the MPC control law. In [21] ellipsoidal feasible invariant sets were computed for a conventional dual mode MPC setup and the implied LTV law was extracted from these. Recently, the extension of these results in [24,25] was used in [188] to specify a more flexible triple mode algorithm, but still for the nominal case. However, as the algorithm in [21,24,25] were originally developed for the robust case, later work [114] proposed a robust triple mode MPC algorithm; this is the base algorithm that will be used for comparisons in this chapter.

Specifically, the intention is to consider the potential benefits of more flexible parameterisation based approaches that have been deployed within dual mode MPC in Section 4.4, 5.3 and 8.3 because it is known that in many cases changing the parameterisation allows substantial improvements in the region of attraction with little or no detriment to performance. The algorithms for selecting the mode 2 dynamics in the original robust triple mode papers [114] are challenging and hence this chapter will consider to what extent a more automated approach of using more flexible function dynamics will be appropriate for the robust case.

The main contributions of this chapter are to extend the more flexible parameterisation

to the triple mode algorithm with comparison to the earlier proposed approaches. This will give a new insight into the potential of the more flexible function parameterisation for enlarging the region of attraction, improving the optimality and computational loads within robust MPC. The contributions of chapter are twofold: Firstly to give a new avenue for the use of more flexible function parameterisation (i.e. Laguerre, Kautz and generalised function) to select the middle mode as an intuitive choice. Secondly, using the insight gained from this to use the generalised function parameterisation to obtain a large region of attraction for both nominal and robust cases.

9.2 Background

This section will summarise the background information related to robust triple mode MPC for convenience (see Section 3.5, 3.7 and 3.4 for further details).

9.2.1 Triple mode MPC algorithm

The triple mode approach to finding the best compromise between a region of attraction and good performance is underexplored in the literature. It is recognised that a large number of free (before resorting to the terminal law) control moves may be required to get close to the global optimal region of attraction whereas one may not desire this large number of degrees of freedom. In triple mode an additional mode is introduced into the predicted class to increase the region of attraction; originally, for computational convenience, this choice was based on the analysis of invariant ellipsoidal sets. However, to form an efficient triple mode algorithm, that is with few optimisation variables, it is necessary to make implicit assumptions for the terminal mode and the middle mode while selecting the initial mode explicitly using polytopic constraints.

In order to select an implicit choice for middle mode, different choices based on ellipsoidal sets with only few parameters were proposed in [113] and [21]. The main idea that is, to define an augmented system model incorporating the mode 1, 2 d.o.f. was proposed in [21] to handle the feasibility maximisation in [24] and [25]. For more details see Section 3.4.

In [114], the triple mode prediction setup is modified in conjunction with GERPC to formulate a robust triple mode MPC algorithm. The proposed algorithm allows a tractable QP-based MPC algorithm for the robust case, it allows a large region of attraction with

just a small number of online optimisation variables. However, further research remaining is to make the algorithm handle non-symmetric constraints better. This chapter proposes flexible function parameterisation to better handle non-symmetric constraints with simplified offline computations.

9.3 Selection of Middle mode within a robust triple Mode MPC

The fundamental weakness of the conventional triple mode algorithm is linked to the efficiency of the middle mode; can this be computed implicitly or explicitly and also is the offline optimisation for identifying a suitable dynamic G_c in (3.57) overly complex? This section explores a more intuitive technique based on predefined dynamics in the middle mode; in this section generalised function dynamic is proposed as these have been shown to be effective within dual mode MPC to enlarge the region of attraction without too much detriment to performance. This section shows how generalised function dynamics is analogous to the mode 2 of GERPC based triple mode and thus can be deployed in the middle mode for a triple mode MPC. This section can equally be reworked for the nominal case.

This section provides an alternative proposal to GERPC [25] to handle non-symmetric constraints better as shown in Section 8.5. It is shown that for the robust case, using generalised function dynamic insights within a robust triple MPC may enlarge the region of attraction and provide a pragmatic choice for selecting the middle mode, thus simplifying offline design by removing the need for demanding optimisations.

9.3.1 Selection of Middle mode using generalised function dynamic

Robust triple mode MPC based on generalised function dynamics introduces a middle mode using generalised functions. The generalised functions give a pragmatic choice for selecting the middle mode within the robust triple mode MPC and without the need for a LMI/BMI optimisation. In terms of model (3.57), the choice for robust triple mode using generalised functions are modified as described below.

The predicted cost (3.61) can be represented in terms of perturbation ρ_k, that is:

$$J_G(x_k, \underset{\rightarrow}{\rho}_k, \underset{\rightarrow}{c}_k) = [x_k \ \underset{\rightarrow}{f}_k \ \underset{\rightarrow}{c}_k]^T P [x_k \ \underset{\rightarrow}{f}_k \ \underset{\rightarrow}{c}_k] \tag{9.1}$$

with $f_{k+i} = G_i^T \underset{\rightarrow}{\rho}_k$ and $G_i = A_G G_{i-1}$. Constraint handling requires a suitable MCAS or $\mathcal{X}_r$ and this is calculated using similar ideas to that in [115], but modified to allow for the triple mode prediction structure, as explained next.

Extension to the robust case requires the use of appropriate invariant sets and inequalities representing constraints which allow for parameter uncertainty. An autonomous prediction formulation, using the generalised function dynamic for the middle mode, is defined as:

$$X_{k+1+i|k} = \Psi_{k+i|k} X_{k+i|k},\ X_{k|k} = \begin{bmatrix} x_k \\ \underset{\rightarrow}{\rho}_k \\ c_k \end{bmatrix}, \tag{9.2}$$

$$\Psi_{k+i|k} \in Co\{\Psi_j,\ j = 1 \ldots, m\},$$

$$[A_G, G_0] \in Co\{[A_{G,j}, G_{0,j}],\ j = 1, \ldots, m\},$$

$$\Psi_j = \begin{bmatrix} \Phi_j & B_j G_0^T & B_j E \\ 0 & A_G^T & 0 \\ 0 & 0 & I_L \end{bmatrix}.$$

All possible evolutions of these dynamics must meet the constraints (3.58), and thus

$$\begin{bmatrix} L_x - L_u K & L_u G_0^T & L_u E \end{bmatrix} X_k \leq l. \tag{9.3}$$

After calculating the robust invariant polyhedral set of (9.2) subject to (9.3) using algorithm 8.1 in [115] as $A_S X_k \leq b_S$ the following generalised function based RTMPC (GRTMPC) algorithm 9.1 is implemented.

Algorithm 9.1. *Generalised function based RTMPC (GRTMPC)*

Off-line

- *Given design parameters $n_c, m, a_1, \ldots, a_m$ and R, calculate G_0^T and A_G from (3.30), where m is order of generalised function dynamics.*
- *Calculate P from* (3.63) *using $f_{k+i} = G_i^T \underset{\rightarrow}{\rho}_k$.*
- *Determine the polyhedral robust control invariant set (i.e. $A_S X \leq b_S$) which can be found for* (9.2) *subject to* (9.3) *using algorithm 8.1.*

On-line

1. *At each sampling instant,*

2. *k=0; if* $x_k \in \mathcal{X}_{0_g}$ (8.19) *implement terminal mode control law i.e.* $u_k = -Kx_k$ *else*

3. *Solve the optimisation problem*

$$\min_{\underset{\rightarrow}{\rho}_k, \underset{\rightarrow}{c}_k} J_G(x_k, \underset{\rightarrow}{\rho}_k, \underset{\rightarrow}{c}_k)$$

$$s.t. \quad A_S X_k \leq b_S.$$

4. *Implement* $u_k = -Kx_k + G_0^T \underset{\rightarrow}{\rho}_k + E \underset{\rightarrow}{c}_k$ *to the plant.*

5. *Set* $k = k + 1$*, repeat until converges.*

6. *End.*

Remark 9.3.1. *Recursive feasibility and robust asymptotic stability of GRTMPC can be proved similarly as in [114].*

Remark 9.3.2. *Generalised function based algorithms can be tuned using a pragmatic selection based on closed loop system poles (see Chapter 6 for further discussion on parameter selection).*

9.3.2 Triple mode MPC or generalised function triple MPC – a comparison

The previous sections have shown that generalised functions are an alternative to GERPC [114] for generating mode 2 dynamics in triple mode MPC algorithms using polytopic constraints. The triple mode MPC algorithm takes dual mode predictions as a base and increases the region of attraction by adding a third mode (in fact what is denoted as mode 2 is the additional mode). The motivation is to enlarge the region of attraction without too much detriment to performance and preferably with little impact on the computational burden.

Triple mode is known to be effective in maximising the region of attraction and without detriment to performance so the key question to discern is whether the strategy is better than just increasing the number of d.o.f. available to a standard RMPC algorithm. Secondly, there is interest in whether the proposed generalised function approach has

benefits over the earlier GERPC based approaches. It should be noted that all cases lead to a quadratic programming (QP) problem - the number of inequalities required to describe the resulting MCAS is investigated.

In terms of offline computations, the proposed approach is a significant improvement on the GERPC based approach: (i) GERPC requires a SDP in order to determine the dynamic G_c whereas (ii) the generalised function algorithm requires only the choice of poles(s) i.e. '$a_j,\ j = 1, 2, \ldots, m$', where in general '$a_j$' improves region of attraction but slows predicted responses and thus this is an intuitive design parameter and a pragmatic choice. Nevertheless GERPC appears more systematic, but because it is based on ellipsoidal rather than the polytopic sets, it may be conservative for non-symmetric constraints. For GERPC, d.o.f. should be chosen equal or greater to system dimension i.e. $m_c \geq n_x$ and it is not known how to specify a convex offline problem when $m_c < n_x$. Therefore, generalised function allow an alternative proposal to design the middle mode when $m_c \leq n_x$.

The pragmatic choice is a suboptimal selection of prediction dynamics using generalised function dynamics but in many cases, as shown in Section 8.5 may enlarge the region of attraction compared to GERPC for both symmetric and non-symmetric constraints. This chapter will focus on comparing the efficacy of the GERPC approach and the generalised function approach within triple mode MPC.

9.4 Robust triple mode MPC using generalised function parameterisation

This section looks at the efficiency of the generalised function parameterisation of the d.o.f. within robust triple mode MPC algorithm. In Chapter 8, it was shown that generalised function parameterisations are an effective alternative to the standard basis set for parameterising the d.o.f. in the prediction set deployed by robust MPC. Specifically it was shown that in many cases changing the parameterisation similar to the nominal case allowed substantial improvements in the region of attraction with a slightly greater number of inequalities.

There are two extensions to formulate an efficient triple mode algorithm using generalised function parameterisation.

- Firstly to define the mode 2 implicitly using an ellipsoidal set while selecting the mode 1 control moves $\underset{\rightarrow}{c}_k$ explicitly using the generalised function parameterisation.

- Secondly to define the mode 2 explicitly using Laguerre or Kautz or higher order function dynamics as a pragmatic choice and the mode 1 control moves $\underset{\rightarrow}{c}_k$ using the generalised function parameterisation.

9.4.1 Generalised function parameterisation based robust triple mode MPC using ellipsoids

The offline problems of ERPC and GERPC can be used to implicitly specify the second mode control moves for LTI and robust triple mode MPC, [113,114,188]. It is tempting to use the first control move of the middle mode control $f_{k+i} = Hx_{n_c}$, where $H = -P_{22}^{-1}P_{21}$. The P_{22} and P_{21} are corresponding to Q_z in (3.55).

As mention earlier (3.55), the Q_z is the matrix defining the augmented invariant ellipsoid $\mathcal{E}_z = \left(z_k : z_k^T Q_z^{-1} z_k \leq 1\right)$, with size to some degree decided by the choice of γ. The $\mathcal{E}_z$ in $z_k = [x_k^T \;\; c_k^T]^T$ can be written as [113, 114]:

$$z_k^T \underbrace{\begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}}_{Q_z^{-1}} z_k \leq 1. \tag{9.4}$$

This fixed control law ensures robust invariance [25,114] and could be used as first control move in a terminal mode in lieu of the optimised feedback $u_k = -Kx_k$.

Hence, define middle mode varying terminal state feedback control law as

$$u_k = (-K + DH)x_k, \tag{9.5}$$

which has the same (robust) region of attraction as (G)ERPC, a dual mode robust MPC algorithm can be constructed by the method [115] using this state feedback as a terminal control (defining a terminal set and cost). The augmented dynamics using generalised function parameterisations are defined as

$$\tilde{\Psi}_j = \begin{bmatrix} A_j - B_j K + DH & B_j E \\ 0 & A_G^T \end{bmatrix} \tag{9.6}$$

where $c_k = G_0^T \underset{\rightarrow}{\rho}_k$ and $\underset{\rightarrow}{\rho}_{k+1} = A_G^T \underset{\rightarrow}{\rho}_k$. These dynamics should fulfill the constraints,

$$\begin{bmatrix} Lx_k - LuK + LuDH & LuG_0^T \end{bmatrix} z_k \leq l. \tag{9.7}$$

A robustly invariant polyhedral set can be found in (9.6) subject to (9.7) using the algorithm 8.1 described in [115], in the form

$$M_r x_k + N_r \underset{\rightarrow}{\rho}_k \leq d_r. \tag{9.8}$$

A predicted cost can be constructed as [114, 188]:

$$J(x_k, \underset{\rightarrow}{\rho}_k) = [x_k \ \underset{\rightarrow}{\rho}_k]^T \tilde{P} [x_k \ \underset{\rightarrow}{\rho}_k] \tag{9.9}$$

where $\tilde{P} > 0$ satisfies

$$\begin{aligned} \tilde{P} - \tilde{\Psi_j}^T \tilde{P} \tilde{\Psi_j} \geq [I \ 0]^T Q [I \ 0] \\ + [-K + DH \ G_0^T]^T R [-K + DH \ G_0^T], \quad j = 1, ..., m. \end{aligned} \tag{9.10}$$

The matrix $\tilde{P}$ can be efficiently calculated by the SDP

$$\min_{\tilde{P}} \ tr(\tilde{P}) \quad s.t. \quad (9.10). \tag{9.11}$$

Although this approach will enlarge the region of attraction over and above a more conventional dual mode approach, even one based on a generalised function parameterisation the size and cost tunable with γ. The online cost will be suboptimal since the cost (for small n_c) will be dominated by the (G)ERPC state feedback designed for maximum stabilisable region. Sub-optimality can be tuned by decreasing γ.

However, this extension reduces the computational complexity by embedding a middle mode with no optimisation variables while still tackling the robust case and thus is a good base for adding an additional mode to give a robust triple mode MPC.

Algorithm 9.2. *Generalised function based robust triple mode MPC using an ellipsoidal set (GR(E)TMPC)*

Off-line

- *Given design parameters $n_c, \gamma, m, a_1, \ldots, a_m, Q$ and R, calculate G_0^T and A_G from* (5.5) *where m is order of generalised function dynamics.*

- *Calculate* $H = -P_{22}^{-1}P_{21}$ *from* (9.4) *and finally calculate* $\tilde{P}$ *from* (9.11).
- *Determine the invariant polyhedral set (i.e.* M_r, N_r *and* d_r*) can be found for* (9.7) *using algorithm 8.1.*

On-line

1. *At each sampling instant,*
2. *k=0; if* $x_k \in \mathcal{X}_{0_g}$*, implement terminal mode control law i.e.* $u_k = -Kx_k$ *else*
3. *Solve the optimisation problem*

$$\min_{\underrightarrow{\rho}_k} [x_k \ \underrightarrow{\rho}_k]^T \tilde{P} [x_k \ \underrightarrow{\rho}_k]$$
$$s.t. \quad M_r x_k + N_r \underrightarrow{\rho}_k \leq d_r.$$

4. *Implement* $u_k = (-K + DH)x_k + G_0^T \underrightarrow{\eta}_k$ *to the plant.*
5. *Set* $k = k + 1$*, repeat until converges.*
6. *End.*

9.4.2 Robust triple mode MPC using generalised function parameterisation

The generalised function parameterisation (i.e. Laguerre, Kautz or higher order dynamic function) may be used to enlarge the region of attraction of robust triple mode MPC. As in the robust dual mode case, the input perturbations c_k are parameterised using generalised functions in (9.2) and the predicted cost can be represented in terms of the perturbations $\tilde{\rho}_k$ in (9.1), hence

$$J_G(x_k, \underrightarrow{\rho}_k, \underrightarrow{c}_k) = [x_k \ \underrightarrow{f}_k \ \underrightarrow{\rho}_k]^T P [x_k \ \underrightarrow{f}_k \ \underrightarrow{\rho}_k], \tag{9.12}$$

with $f_{k+i} = G_i^T \underrightarrow{\rho}_k$, $c_{k+i} = \tilde{G}_i^T \underrightarrow{\tilde{\rho}}_k$, $G_i = A_G G_{i-1}$ and $\tilde{G}_i = \tilde{A}_G \tilde{G}_{i-1}$. Different generalised function dynamics can be used for middle mode and parameterisation of input perturbations.

From (9.2), the triple mode augmented dynamics can be modified by replacing I_L by $\tilde{A_G}$ where

$$[\tilde{A_G}, \tilde{G_0}] \in Co\{[\tilde{G}, j, \tilde{G}_{0,j}],\ j = 1, \ldots, m\}. \tag{9.13}$$

These dynamics should fulfill the constraints,

$$\begin{bmatrix} L_x - L_u K & L_u G_0^T & L_u \tilde{G}_0^T \end{bmatrix} X_k \leq l. \tag{9.14}$$

Robust constraint handling is represented by an MCAS or $\mathcal{X}_r$ which is calculated offline with the methodology of [83, 115], but deploying alternative functions, that is equations (9.3), (9.13, 9.14) within the updated model; this is illustrated in the algorithm 8.1. After calculating the inequalities which describe the invariant set the GRTMPC algorithm 9.1 is equally reworkable using generalised function parameterisation in (9.12), (9.13) and (9.14).

For completeness define the MCAS for GRTMPC as

$$\mathcal{X}_{r_g} = \{x : \exists(\underset{\rightarrow}{\rho}_k, \underset{\rightarrow}{\tilde{\rho}}_k)\ s.t.\ A_S X \leq b_S,\ X = [x, \underset{\rightarrow}{\rho}_k, \underset{\rightarrow}{\tilde{\rho}}_k]^T\}, \tag{9.15}$$

and associated MAS is defined as

$$\mathcal{X}_{0_g} = \{x : \exists(\underset{\rightarrow}{\rho}_k, \underset{\rightarrow}{\tilde{\rho}}_k) = (0, 0)\ s.t.\ A_S X \leq b_S,\ X = [x, 0, 0]^T\}. \tag{9.16}$$

Remark 9.4.1. *Although not discussed here to avoid tedious but straightforward algebra, the Laguerre and Kautz dynamics equally reworkable for all proposed algorithms as a special case of generalised function.*

Remark 9.4.2. *All these algorithms can equally be reworked for the nominal case.*

9.4.3 Selection of Middle mode using generalised function dynamic

In robust triple mode MPC, it was noted [23, 24] that much larger ellipsoids could be obtained for smaller m_c in (3.56) if one allowed dynamics in the predictions, but at the cost of a non-convex (Bilinear Matrix Inequality - BMI) offline problem. Shortly after [25], it was shown that if $m_c \geq n_x$, it is possible to specify an equivalent convex semi-definite programming problem, and moreover, that in terms of the size of regions of attraction, there is no advantage in choosing $m_c > n_x$.

The generalised dynamics can be selected with $m_c \leq n_x$ and the region of attraction enlarges further using the generalised function parameterisation in (9.13). The generalised function dynamics allow a pragmatic selection for the middle mode when $m_c \leq n_x$, whereas, in GERPC it is not known how to specify a convex offline problem when $m_c < n_x$. The generalised function parameterisation can be used to further enlarge the region of attraction with a pragmatic selection of middle mode using generalised function dynamics and even with a systematic selection of middle mode using GERPC.

Theorem 9.1. *The GRTMPC algorithm has a guarantee of stability and recursive feasibility.*

Proof. Recursive feasibility: Given an optimal $\underset{\rightarrow}{c}_k = [G_0^T, \ldots, G_{n_c-1}^T]^T \underset{\rightarrow}{\rho}_k$ at time k, it is clear from the key requirement of GRTMPC algorithm that is

$$x_k \in \mathcal{X}_{r_g} \implies x_{k+1} \in \mathcal{X}_{r_g} \tag{9.17}$$

and in fact to be more precise, one requires that an augmented state including the tail of $\underset{\rightarrow}{c}_k = [G_0^T, \ldots, G_{n_c-1}^T]^T \underset{\rightarrow}{\rho}_k$ remains feasible, that is $\underset{\rightarrow}{\rho}_{tail|k} = [\rho_{k+1|k}^T, \ldots, \rho_{k+n_c-1|k}^T, 0]^T$ is a feasible first mode control sequence at time $k+1$. Repeating this argument proves recursive feasibility. Such a guarantee can be established by defining a robust invariant set using algorithm 8.1 using [115].

Asymptotic stability: Asymptotic stability has followed since $J_{G,k} := J_G(x_k, \underset{\rightarrow}{\tilde{\rho}}_k, \underset{\rightarrow}{\rho}_k)$ is positive definite and monotonically non-increasing (thus it is a Lyapunov function): Using the optimal input at time k to construct a feasible input for time $k+1$ using recursive feasibility, it is clear from [6] that $J_{G,k+1} \leq J_G(x_{k+1}, \underset{\rightarrow}{\tilde{\rho}}_{k+1}, [\rho_{k+1|k}^T, \ldots, \rho_{k+n_c-1|k}^T, 0]^T) < J_{G,k}$ for $x_k \neq 0$. □

9.5 Numerical Examples

This section will illustrate the efficacy of the parameterisation within robust triple mode MPC algorithms by numerical examples given next. The aim is to compare two aspects: (i) the size of the regions of attraction; (ii) the number of inequalities required to describe the robust MCAS. For the purposes of visualization, figures are restricted to second order system for which it is possible to plot the regions of attraction. The robust and nominal

cases are simulated using both symmetric and non-symmetric constraints. The nominal dynamics are given by $A = 0.5(A_1 + A_2)$ and $B = 0.5(B_1 + B_2)$.

Example 1
Consider a linear uncertain system representing a double integrator with an uncertainty polytope are defined by the following two vertices:

$$A = \begin{bmatrix} 1 & \zeta_1 \\ 0 & 1 \end{bmatrix};\ B = \begin{bmatrix} 0 \\ \zeta_2 \end{bmatrix}; \zeta_1 = (0.1, 0.2),$$
$$\zeta_2 = (1, 1.5)\ Q = I, R = 0.1, m_c = 2, \gamma = 10^{10}. \tag{9.18}$$

The system is subject to input and state symmetric constraints

$$-1 \leq u_k \leq 1; \qquad -10 \leq x_{i_k} \leq 10;\ i = 1, 2; \tag{9.19}$$

and non-symmetric constraints

$$-1.5 \leq u_k \leq 1; \qquad -15 \leq x_{i_k} \leq 10;\ i = 1, 2; \tag{9.20}$$

The alternative dynamics i.e. Laguerre and Kautz dynamics are selected in the vicinity of closed loop stable pole(s). These dynamics are selected as a combination to define both mode 1 and 2 respectively. Laguerre dynamics are $p = (0.6, 0.7)$ and Kautz dynamics are $(a, b) = ((0.7, 0.1), (0.6, 0.1))$.

Example 2
Consider another 2nd order (i.e. $x \in \mathbb{R}^2$) linear uncertain system representing an uncertainty polytope are defined by the following two vertices:

$$A = \begin{bmatrix} 0.6 & \zeta_1 \\ 1 & 1.4 \end{bmatrix};\ B = \begin{bmatrix} 0.2 \\ \zeta_2 \end{bmatrix}; C = \begin{bmatrix} 1 & 0 \end{bmatrix}; \zeta_1 = (-0.4, -0.5),$$
$$\zeta_2 = (0.05, 0.5)\ Q = C^T C, R = 1, \gamma = 10^{10}, m_c = 2. \tag{9.21}$$

The system is subject to input and state symmetric constraints

$$-1 \leq u_k \leq 1; \qquad -10 \leq x_{i_k} \leq 10;\ i = 1, 2; \tag{9.22}$$

and non symmetric constraints

$$-1.5 \leq u_k \leq 1; \qquad -15 \leq x_{i_k} \leq 10;\ i = 1, 2; \tag{9.23}$$

The alternative dynamics i.e. Laguerre and Kautz dynamics are selected similarly as in the previous example: Laguerre dynamics are $p = 0.6$ and Kautz dynamics are $(a, b) = (0.6, 0.1)$.

Example 3
Consider a 3rd order (i.e. $x \in \mathbb{R}^3$) linear uncertain system representing an uncertainty polytope are defined by the following two vertices:

$$\begin{aligned} A_1 &= \begin{bmatrix} \zeta_1 & -\zeta_2 & -0.1080 \\ 2 & 0 & 0 \\ 0 & \zeta_3 & 0 \end{bmatrix};\ B_1 = \begin{bmatrix} \zeta_4 \\ 0 \\ 0 \end{bmatrix}; \\ &\zeta_1 = (1.4, 1.5), \zeta_2 = (0.105, 0.205), \zeta_3 = (1, 1.5), \zeta_4 = (0.2, 0.3); \\ &Q = I, R = 1, \gamma = 10^{10}, m_c = 3. \end{aligned} \tag{9.24}$$

The system is subject to input and state symmetric constraints

$$-1 \leq u_k \leq 1; \qquad -1 \leq x_{i_k} \leq 1;\ i = 1, 2, 3; \tag{9.25}$$

and non-symmetric constraints

$$-0.5 \leq u_k \leq 1; \qquad -1.5 \leq x_{i_k} \leq 1;\ i = 1, 2, 3; \tag{9.26}$$

The alternative dynamics are selected as: Laguerre dynamics are $p = 0.5$, Kautz dynamics are $(a, b) = (0.5, 0.1)$ and $3rd$ order generalised function dynamics are $(a_1, a_2, a_3) = (0.5, 0.1, 0.1)$.

9.5.1 Robust triple mode MPC – using explicit selection of middle mode

The middle mode within the robust triple mode MPC is introduced using GERPC (RTMPC), Laguerre (LRTMPC), Kautz (KRTMPC)and generalised function dynamics (GRTMPC). The selection of generalised function dynamics is made in line with $m_c \leq n_x$.

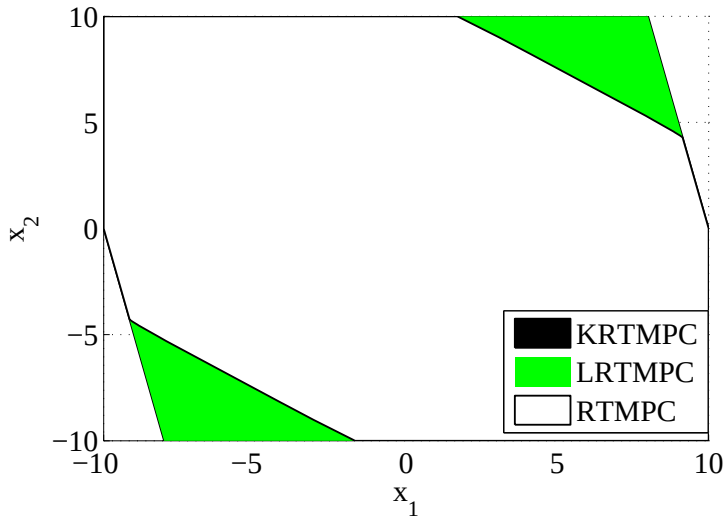

Figure 9.1: Regions of attraction for model (9.18) using Robust triple mode MPC, $n_c = 2$, $m_c = 2$, $\gamma = 10^{10}$ – for symmetric constraints

The simulations are done using middle mode same as the dual mode MPC and using the d.o.f.. The d.o.f. within the robust triple MPC is introduced using ERPC, Laguerre and Kautz function parameterisation for Algorithm 9.1. The results for all examples using both symmetric and non-symmetric constraints are presented in Table 9.1. The d.o.f. shown in Table 9.1 is the sum of m_c and n_c.

In Example 1, Figure 9.1, and 9.2 show the regions of attraction using GERPC, Kautz and Laguerre function dynamics for $m_c = 2$ and $n_c = 2$ for uncertain case, whereas, the region of attraction using $m_c = 2$ are shown in Figure 9.3, and 9.4 for symmetric and non-symmetric constraints respectively. The region of attraction for both Laguerre and Kautz dynamics are similar in volume and larger than GERPC, utilising the different number of inequalities as shown in Table 9.1.

For Example 2, Figure 9.5 and 9.6 show the region of attraction for symmetric and non-symmetric constraints respectively using $m_c = 2$ and $n_c = 2$ for the uncertain case. The Kautz function dynamic has a larger region of attraction than Laguerre and GERPC dynamics as shown in Table 9.1 for both symmetric and non-symmetric constraints.

For Example 3, the middle mode dynamics are introduced using $m_c = 3$ for GERPC, Laguerre, Kautz and $3rd$ order generalised function dynamics and the d.o.f. in mode 1 is parameterised using Laguerre function dynamics for $n_c = 1$. Laguerre, Kautz and generalised function dynamics enlarge the region of attraction compared to GERPC for both symmetric and non-symmetric constraints, whereas, Laguerre, Kautz and $3rd$ order dynamics have similar MCAS volume with different number of inequalities as shown

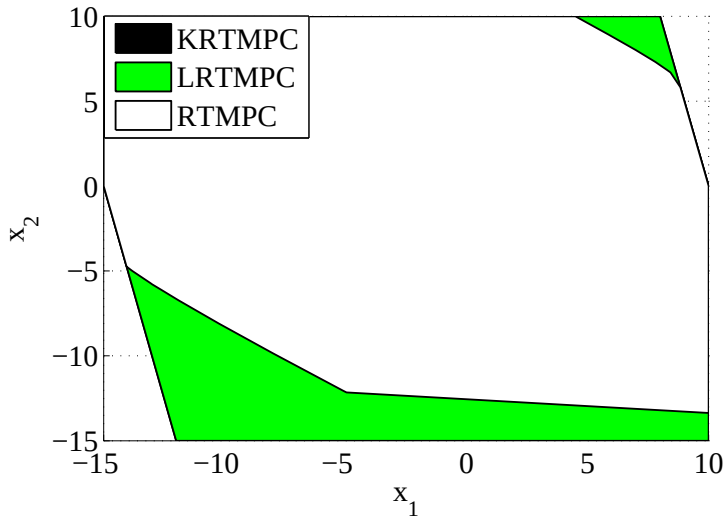

Figure 9.2: Regions of attraction for model (9.18) using Robust triple mode MPC, $n_c = 2$, $m_c = 2$, $\gamma = 10^{10}$ – for non-symmetric constraints

in Table 9.1. The Kautz and $3rd$ order dynamics have fewer inequalities compared to Laguerre function. The higher order dynamics with similar region of attraction can be used to simplify the computational burden with fewer inequalities as shown in Table 9.1.

The Laguerre, Kautz and generalised function dynamics used within the robust triple mode MPC enlarge the region of attraction for both symmetric and non-symmetric constraints. From the simulation results, the selection of algorithms based solely on MCAS volume indicated that the alternative parameterisation with dynamic equal to system dimension is preferred in general.

9.5.2 Nominal triple mode MPC – using explicit selection of middle mode

In nominal cases, the simulation results for all examples using symmetric and non-symmetric constraints are shown in Table 9.2. Similar to the robust case, Laguerre, Kautz and generalised function enlarge the region of attraction compared to GERPC for $2nd$ order dynamic systems. For Example 3, TMPC (using GERPC), LTMPC (using Laguerre dynamics), KTMPC (using Kautz dynamics) and GTMPC (using generalised dynamics) has the similar region of attraction with different number of inequalities, whereas, GTMPC with $3rd$ order generalised function dynamic has fewer inequalities than the other algorithms.

Similarly for the robust case, the generalised function dynamics equal to the system

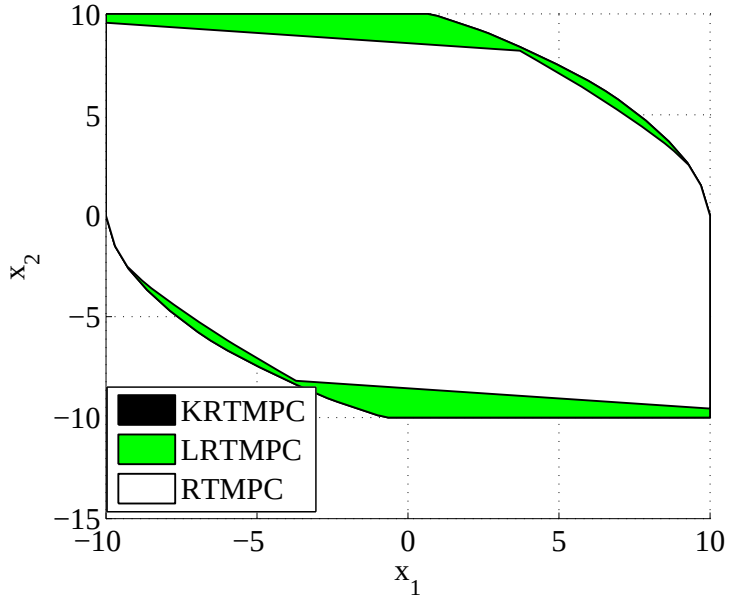

Figure 9.3: Regions of attraction for model (9.18) using Robust triple mode MPC, $m_c = 2$, $\gamma = 10^{10}$ – for symmetric constraints

dimension are preferred to enlarge the region of attraction with simplified computational burden.

9.5.3 Robust triple mode MPC – using implicit selection of middle mode

The ellipsoidal based approaches to define middle mode varying terminal control law (i.e. $u_k = (-K + DH)x_k$) enlarge the region of attraction using generalised function parameterisation. The middle mode dynamics are implicit within the prediction and not present in the prediction set as f_k only tend to zero asymptotically [188]. The simulation results are shown in Table 9.3 and 9.4 for uncertain and nominal cases respectively.

In Example 1, Laguerre and Kautz function parameterisation enlarge the region of attraction significantly using both symmetric and non-symmetric constraints compared to ERPC using $n_c = 2$. For Example 2 and 3, there is not much difference between the volume of MCAS using parameterised (i.e. Laguerre and Kautz function) compared to ERPC using $n_c = 2$, whereas in Example 3, the MCAS volume is similar for all algorithms i.e. LR(E)TMPC (using Laguerre dynamics), KR(E)TMPC (using Kautz dynamics), GR(E)TMPC (using generalised dynamics) and R(E)TMPC (using GERPC) using $n_c = 3$ with a different number of inequalities.

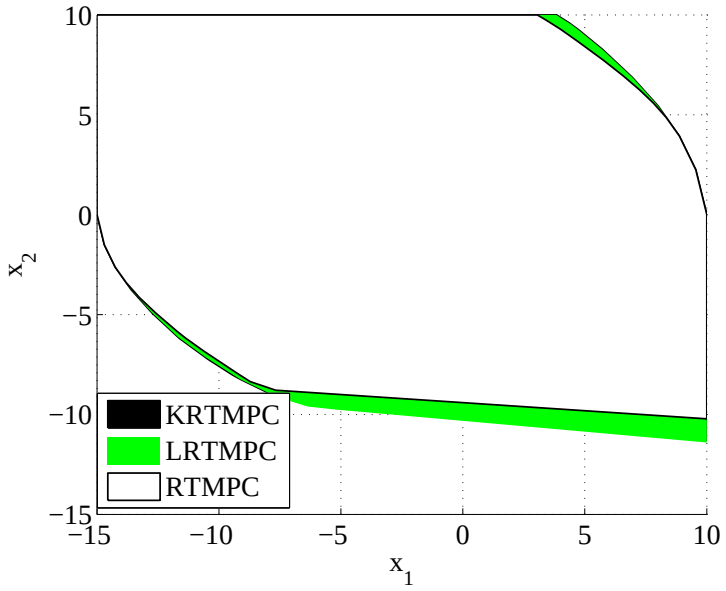

Figure 9.4: Regions of attraction for model (9.18) using Robust triple mode MPC, $m_c = 2$, $\gamma = 10^{10}$ – for non-symmetric constraints

The parameterised algorithms enlarge the region of attraction at the price of an increase in the number of inequalities as shown in Table 9.3. It is interesting to observe from the Table 9.3 that the Kautz function parameterisation has fewer inequalities than Laguerre and generalised function parameterisation algorithms.

9.5.4 Nominal triple mode MPC – using implicit selection of middle mode

The nominal triple mode MPC with the middle mode varying terminal control is simulated using symmetric and non-symmetric constraints. The simulation results are compared in Table 9.4 using MCAS volume and number of inequalities.

In Example 1, similarly to the uncertain cases, Laguerre and Kautz function parameterisation enlarge the region of attraction with a slight increase in the number of inequalities compared to ERPC using $n_c = 2$, whereas in Example 2, there is slightly improvement in the MCAS volume using parameterised algorithms compared to ERPC for both symmetric and non-symmetric constraints.

In Example 3, the simulation results are compared using both $n_c = 2$ and $n_c = 3$. For $n_c = 2$, there is a slight enlargement in the region of attraction using parameterised algorithm, whereas for $n_c = 3$, all algorithms have the similar MCAS volume with dif-

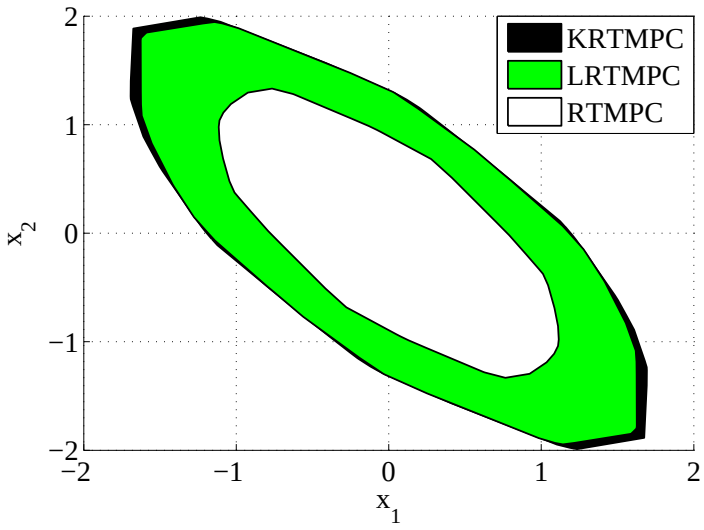

Figure 9.5: Regions of attraction for model (9.21) using Robust triple mode MPC, $n_c = 2$, $m_c = 2$, $\gamma = 10^{10}$ – for symmetric constraints

ferent number of inequalities. The number of equalities using $3rd$ order function parameterisation is similar to the ERPC algorithm, so in this case ERPC is preferred over parameterised algorithms.

9.5.5 Closed loop performance

A fair comparison of closed loop performance would require the same initial conditions for all algorithms. To demonstrate how the proposed algorithm performs, the closed loop performance of KRTMPC, LRTMPC, GERPC and ERPC is contrasted for 200 initial conditions near the boundary of the region of attraction of ERPC. Figure 9.7 shows the region of attractions and 200 feasible initial conditions which are contrasted for symmetric constraints using (9.1) and (9.12).

For Example 1, Table 9.5 represents an average cost comparison for both nominal and robust cases. LRTMPC enlarge the region of attraction with insignificant performance degradation as compared to ERPC for nominal case, whereas an improvement in the performance is obtained for robust cases. KRTMPC has an improvement in the closed loop performance despite having a similar region of attraction as compared to LOMPC for both nominal and robust cases.

Table 9.5 shows that in both cases, ERPC significantly outperforms GERPC for $\gamma = 10^{10}$. Tuning GERPC for better performance (reducing γ) gives small region of attraction, but

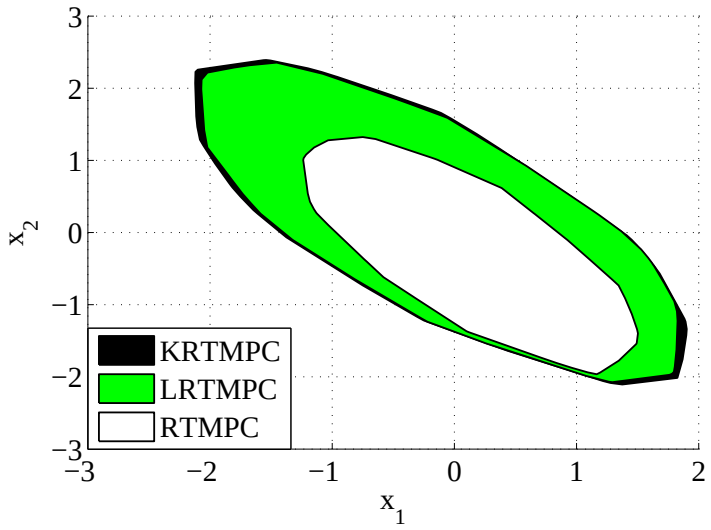

Figure 9.6: Regions of attraction for model (9.21) using Robust triple mode MPC, $n_c = 2$, $m_c = 2$, $\gamma = 10^{10}$ – for non-symmetric constraints

the cost gets closer to the ERPC cost. KRTMPC and LRTMPC enlarge the region of attraction as compared to GERPC (as shown in Figure 9.7) without insignificant performance degradation.

9.5.6 Computational complexity

For completeness, it is important to compare the number of inequalities required to describe the robust MCAS as the complexity of these set descriptions has an impact on the online computational burden, the more inequalities the higher the computational burden in solving the associated QP optimisation (this chapter does not discuss issues linked to the exploitation of structure and efficient QP optimisers). The number of inequalities to define the MCAS is compared with the number of d.o.f. in Table 9.1, 9.2, 9.3 and 9.4.

The alternative algorithms (i.e. using Laguerre, Kautz and generalised function dynamics) enlarge the MCAS volume at the price of an increase in the number of constraints in the online problem using an implicit or an explicit choice of triple mode MPC. The higher order function dynamics can be used to reduce the number of inequalities, but it cannot be proved generally.

Table 9.1: Comparison of MCAS volume and number of inequalities for Robust triple mode MPC

	Symmetric constraints								
	Example 1			Example 2			Example 3		
Algorithm	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.
RTMPC	344.94	227	4	3.18	144	4	1.75	78	4
	307.39	36	2	2.62	92	2	1.75	76	3
LRTMPC	380.00	130	4	6.87	190	4	1.78	150	4
	338.24	88	2	2.64	98	2	1.75	142	3
KRTMPC	380.00	130	4	7.06	178	4	1.78	104	4
	338.24	34	2	2.67	98	2	1.75	71	3
GRTMPC	-	-	-	-	-	-	1.78	104	4
	-	-	-	-	-	-	1.75	71	3
	Non-symmetric constraints								
	Example 1			Example 2			Example 3		
Algorithm	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.
RTMPC	505.44	140	4	4.67	166	4	3.10	97	4
	436.68	65	2	3.85	115	2	3.01	77	3
LRTMPC	592.50	122	4	9.15	198	4	3.47	138	4
	456.22	89	2	3.87	111	2	3.01	76	3
KRTMPC	592.50	113	4	9.42	207	4	3.47	76	4
	453.72	37	2	3.88	124	2	3.01	70	3
GRTMPC	-	-	-	-	-	-	3.47	76	4
	-	-	-	-	-	-	3.01	67	3

9.6 Conclusion

The main contribution of this chapter is to present the applicability of the generalised functions to triple mode MPC. The generalised functions are embedded within the middle mode of both nominal and robust triple mode MPC. These provide a pragmatic choice to enlarge the region of attraction which simplifies the offline design. The generalised function also used to parameterise the degree of freedom within the triple mode MPC. The examples demonstrate that in many cases such a parameterisation may improve the robust region of attraction but possibly with an increase in number of inequalities required to describe the corresponding robust MCAS compared to a more conventional robust approach. Consequently, the use of the generalised function parameterisations within a robust triple mode MPC provides an avenue worth pursuing further.

Table 9.2: Comparison of MCAS volume and number of inequalities for nominal triple mode MPC

	Symmetric constraints								
	Example 1			Example 2			Example 3		
Algorithm	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.
TMPC	351.34	62	4	12.92	22	4	2.50	22	4
LTMPC	380.99	110	4	364.83	28	4	2.50	28	4
KTMPC	385.00	32	4	400	22	4	2.50	20	4
GTMPC	-	-	-	-	-	-	2.50	20	4
	Non-symmetric constraints								
	Example 1			Example 2			Example 3		
Algorithm	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.
TMPC	593.48	72	4	21.03	23	4	4.37	21	4
LTMPC	600.62	32	4	532.02	29	4	4.37	27	4
KTMPC	600.62	28	4	537.94	21	4	4.37	22	4
GTMPC	-	-	-	-	-	-	4.37	20	4

Table 9.3: Comparison of MCAS volume and number of inequalities for Generalised function based robust triple mode MPC using an ellipsoidal set (GR(E)TMPC)

	Symmetric constraints								
	Example 1			Example 2			Example 3		
Algorithm	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.
R(E)TMPC	152.21	24	2	2.77	58	2	1.77	28	2
							1.78	56	3
LR(E)TMPC	359.08	50	2	3.1625	88	2	1.78	30	2
							1.78	70	3
KR(E)TMPC	366.48	40	2	3.19	74	2	1.78	22	2
							1.78	32	3
GR(E)TMPC	-	-	-	-	-	-	1.78	40	3
	Non-symmetric constraints								
	Example 1			Example 2			Example 3		
Algorithm	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.
R(E)TMPC	237.54	23	2	3.67	67	2	3.44	25	2
							3.47	48	3
LR(E)TMPC	557.61	50	2	4.65	93	2	3.47	30	2
							3.47	68	3
KR(E)TMPC	565.82	40	2	4.67	90	2	3.47	24	2
							3.47	34	3
GR(E)TMPC	-	-	-	-	-	-	3.47	54	3

Table 9.4: Comparison of MCAS volume and number of inequalities for nominal G(E)TMPC

	Symmetric constraints								
	Example 1			Example 2			Example 3		
Algorithm	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.
(E)TMPC	194.83	14	2	6.70	16	2	2.53	16	2
							2.56	18	3
L(E)TMPC	367.92	28	2	8.92	26	2	2.55	18	2
							2.56	24	3
K(E)TMPC	374.83	24	2	9.03	24	2	2.56	16	2
							2.56	18	3
G(E)TMPC	-	-	-	-	-	-	2.56	18	3
	Non-symmetric constraints								
	Example 1			Example 2			Example 3		
Algorithm	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.	vol.	Ineq.	d.o.f.
(E)TMPC	304.01	14	2	3.67	67	2	4.95	15	2
							4.99	19	3
L(E)TMPC	572.17	29	2	4.65	93	2	4.99	19	2
							5	27	3
K(E)TMPC	582.39	24	2	4.67	90	2	5	16	2
							5	20	3
G(E)TMPC	-	-	-	-	-	-	5	19	3

Table 9.5: Comparison of average runtime cost for Example 1 using $n_c = 2$

		Robust triple MPC			
Constraints	ERPC	GERPC $\gamma = 10^2$	GERPC $\gamma = 10^{10}$	LRTMPC	KRTMPC
Symmetric	267.25	367.47	432.25	263.98	251.11
Non-symmetric	315.29	435.73	514.44	311.58	296.04
		Nominal triple MPC			
Constraints	ERPC	GERPC $\gamma = 10^2$	GERPC $\gamma = 10^{10}$	LTMPC	KTMPC
Symmetric	263.62	281.34	431.70	271.76	257.34
Non-symmetric	313.14	334.52	511.01	320.28	305.35

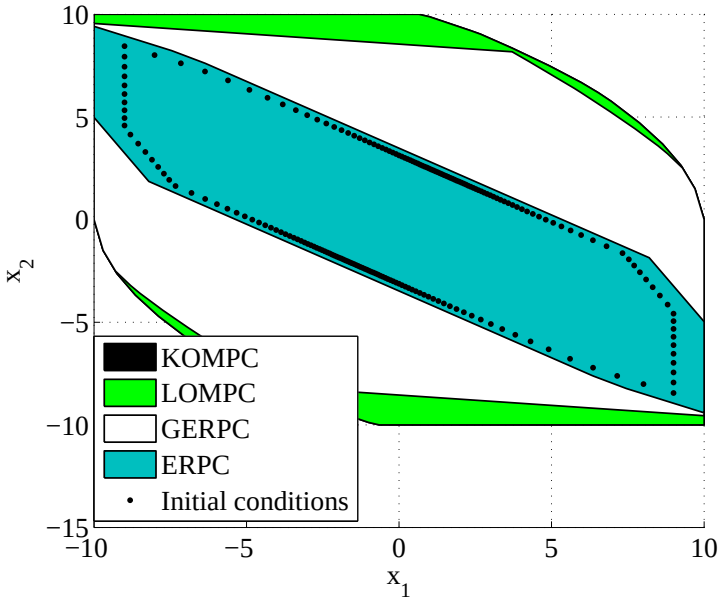

Figure 9.7: Regions of attraction for model (9.18) using Robust triple mode MPC, $m_c = 2$, $\gamma = 10^{10}$ – for symmetric constraints

Part IV

Conclusions and Future Directions

Chapter 10

Conclusions and Future Work

This final chapter is organised as follows: Section 10.1 presents the conclusion of the proposed work in this thesis. This is followed by the summary of the original contributions in Section 10.2 and finishes with the proposed future work in Section 10.3.

10.1 Conclusions

MPC approaches have the advantage of naturally handling multivariable control problems and systems with complex dynamics. These approaches are powerful and robust (more than the standard PID control), and easy to configure and tune. During the past three decades, MPC has proved enormously successful in the industry, mainly because it addresses the constraints in an systematic way. There are several successful theoretical approaches but few of them are implemented commercially by the real time implementation. One important issue for real time implementation is to solve an optimisation problem within a time determined by the sampling interval of the application and therefore computational efficiency of an algorithm becomes critical. A trade off has to be made between performance, region of attraction and the computational burden when choosing from the currently available algorithms. Due to the computationally expensive online optimisation which is required, there has been some limitation to which processes MPC has been used on.

Although computational speed and optimisation algorithms are continuously improving, traditionally such solvers have only been able to handle relatively low control input update rates. However, explicit solutions for the constrained MPC problem formulation

significantly increase the potential application area with low dimensional systems. Therefore conventional MPC applications have been limited to situations which, in some sense, justify the cost of such hardware and software and which allow a sufficient time span for solving the overall optimisation problem. Moreover, the implementations via real time solvers is not well suited for all situations which require portable and/or embedded control devices. Thus, there is a requirement to propose an algorithm which simplifies the trade off between performance, region of attraction, and optimisation complexity. This thesis explored alternative parameterisations to simplify this trade off and thus increase the commercial implementation of MPC algorithms.

The first part of the thesis provides the theoretical foundations and a brief historical perspective of MPC algorithms. Thereafter, it briefly discussed nonlinear MPC with implementation limitations. Lack of robustness may compromise the performance of real time implementations of MPC algorithms. A detailed overview of robustness within MPC was presented. In real time implementation, two popular ways to implement the MPC algorithms are explicit and implicit solutions of the finite horizon optimal control problem. An overview of both explicit and implicit solutions was given with real time implementation limitations. Subsequently, a quick overview of efficient algorithms along with challenges for real time implementations was given in Chapter 2. After that Chapter 3 provided a common theoretical background necessary for arguments in this thesis. It introduced a problem formulation, stability within the dual mode prediction, Laguerre function parameterisation, and triple mode approaches using ellipsoidal and polytopic sets. Thereafter robust MPC was formulated for both dual mode and triple mode approaches using a linear parameter varying system. Finally, it was highlighted that there is a well understood trade off between region of attraction, performance and inexpensive optimisation.

The second part of the thesis proposed an alternative parameterisation technique using orthonormal basis functions for shaping the predicted input trajectories within MPC and hence to give a more general class. The first contribution was to present the Kautz function as an alternative way to parameterise the input predictions in dual mode MPC. It was shown that Laguerre functions are a special case of Kautz functions. Laguerre functions are generated from a 1st order dynamic (i.e. with a single pole) whereas Kautz functions are generated from 2nd order dynamics (i.e. with two poles). It was shown through examples (refer to Figure 4.4, 4.7, 4.9, 4.11 and Table 4.1) that feasibility can be improved without too much degradation of the performance. However, of more significance, Chapter 4 has tackled the question concerning the earlier proposed use of Laguerre functions to parameterise the d.o.f. in the predictions and clearly demonstrated that obvious alternatives do exist.

The potential benefits of more flexible parameterisations were further explored in Chapter 5 to give a more general class of more flexible function parameterisation. The mathematical representation of the generalisation network was formulated using a state space model. Laguerre and Kautz functions were presented as a special case of generalised functions. The proposed general class was aligned with the standard basis set for an optimal MPC that can be formulated using the generalised function dynamics having all poles at the origin (i.e. all poles equal to zero). It has been shown through numerical examples (refer to Figure 5.1, 5.2, 5.3 and 5.4; and Table 5.3) that in many cases the generalised function parameterisation may give significant feasibility benefits without too much detriment to closed loop performance and while facilitating much lower dimensional optimisations that is possible with a standard optimal MPC approach. While this benefit cannot be proven generally and in some cases is small, there is sufficient evidence to encourage users to try this out as, at times, the benefits can be significant.

GOMPC as a special case of GERPC in [25] gives the most flexibility in the shapes of the input predictions then KOMPC, LOMPC and OMPC, but at the price of a more involved prediction structure. Where less flexibility is required, a designer may choose to use $KOMPC$, $LOMPC$ or even $OMPC$. The key point is that this suite of parameterisations offers a systematic path to follow when for example, OMPC is not giving adequate feasibility for reasonable values of d.o.f. or n_c. In Chapter 6, two systematic techniques were proposed for selecting the parameterisation dynamics based on optimal selection and a pragmatic approach based on stable closed loop dynamics. An optimisation selection was proposed using a multiobjective optimisation based on trade off curves between MCAS volumes, average performance and number of d.o.f.. It is recognised that multiobjective optimisations can be very demanding, albeit offline. So, although these offer good insight into the trade offs and thus what can be achieved, it may not be a useful tool for the average engineer who wants more simplistic but effective design guidance. A pragmatic and simple selection method was demonstrated to identify the parameterised dynamics which, albeit with sub optimal parameter value(s). Further in Chapter 7, the computational analysis was done for both implicit and explicit solutions using generalised function parameterisation. It was shown that in the case of the same number of d.o.f. for both explicit and implicit implementations, then one can find that OMPC may still be competitive in terms of computational load but the feasibility is severely restricted. In contrast, for the case of the global region of attraction, and using as many d.o.f. as are required, then generalised function parameterisation approaches in many cases were computationally efficient. It was concluded in the case of linear time invariant system using a systematic selection of proposed generalised function parameterisation may simplify the trade off within MPC algorithms.

The third part extends the efficacy of generalised function parameterisation to robust MPC algorithms. There is a typical trade off between computational burden and region of attraction. In Chapter 8, generalised function parameterisation was proposed to enlarge the region of attraction while tackling the robust case, perhaps at some small loss of optimality and without compromising the computational burden. Different flexible function parameterisation including Laguerre, Kautz and higher order functions were embedded within the robust MPC approach. A modified augmented formulation was proposed to compute a robust control invariant set, which guaranteed recursive feasibility and convergence. Numerical examples demonstrate (refer to Table 8.1) that in many cases where such parameterisation may enlarge the region of attraction, although with a slight increase in the number of inequalities within a dual mode paradigm. The more flexible function parameterisation was further extended in Chapter 9 to the triple mode paradigm and showed similar benefits accrue. In triple mode approaches, the main motivation was to enlarge the region of attraction without detriment to performance and preferably with little impact on the computational burden. The main weakness of the conventional triple mode approaches was linked to the efficiency of the middle mode; can this be computed implicitly or explicitly and offline optimisation for identifying a suitable dynamic for middle mode may be overly complex. Laguerre and Kautz functions were embedded and evaluated in the middle of both nominal and robust scenarios. These provided a pragmatic choice to enlarge the region of attraction and simplifying the offline computation. A combination of Laguerre and Kautz functions were utilised to define the middle mode as well as parameterising the d.o.f.. The numerical examples demonstrate (refer to Table 9.1 and 9.3) that in many cases such a parameterisation may enlarge the region of attraction but possibly with an increase in number of inequalities required to describe the corresponding robust maximal admissible set compared to a more conventional robust approaches.

As a final remark this thesis proposed a general class of parameterisation functions to simplify the trade off between region of attraction, performance and computational burden consider for both nominal and uncertain cases. The higher order function parameterisation provides more flexibility to shape the input predictions, but at the price of increase in number of d.o.f.. As there is a direct relationship between the order of generalised function dynamics and d.o.f. or n_c within the MPC formulation. A systematic mechanism was proposed for a designer to choose the generalised function dynamics or even OMPC. The key point for the designer is to follow a systematic path to overcome the trade off within the MPC algorithm. In this thesis the underlying MPC problem formulation was considered and proposed generalised function parameterisation to enhance the real time implementation using the advance implementation techniques.

10.2 Original Contributions

The novelty and original contribution of the work presented in this thesis is to propose efficiently parameterised solutions of predictive control. The specific contributions in this work can be summarised as follows:

★ A novel parameterisation for the input sequences in optimal predictive control was proposed. An improvement of the region of attraction of the algorithm was achieved when Kautz functions were used in combination with LOMPC and OMPC without too much detriment to performance and retaining fundamental properties of the OMPC algorithm such as stability and recursive feasibility. It was further explored and a general class of function parameterisation was proposed. A generalised function based MPC algorithm was formulated with guaranteed convergence and recursive feasibility. It was also shown that the OMPC algorithm can be formulated using general class function parameterisation with all dynamic poles place at the origin. The generalised function parameterisation accrues benefits without increasing n_c which simplifies the computational burden.

★ Two novel techniques for selecting the parameterisation dynamics of the general class were proposed based on multiobjective optimisation and a pragmatic choice based on a stable terminal mode control law. These approaches provided a good insight into the choices available. The multiobjective optimisation was formulated using trade off between region of attraction, performance and the number of d.o.f.. Where such an offline analysis is not realistic in some applications, a pragmatic and simple selection method was demonstrated to be effective.

★ The generalised functions were used to parameterise the input sequences in OMPC to achieve an approximately global region of attraction, there was reduction in number of inequalities to represent the region of attraction, the number of regions (and therefore computational complexity and memory storage) using multiparametric QP and also the computational time using active set methods.

★ The parameterised solutions were extended to robust MPC algorithms. A robust control invariant set was proposed using the generalised function parameterisation. The generalised function parameterisation was embedded within the dual mode robust MPC approach using the proposed robust invariant set. The proposed algorithm provided guaranteed convergence and recursive feasibility. It was shown that the proposed algorithm in many cases may improve the region of attraction without any significant change to the number of inequalities required to describe the robust control invariant set which simplifies the online computations.

★ Finally, the parameterised solution was extended to triple mode approaches to simplify the offline computations. The first novelty was to propose explicit choices of the middle mode using generalised functions as a pragmatic choice without demanding offline computations. The second novel contribution was to parameterise the input sequences for both explicit and implicit choices of the middle mode within triple mode MPC algorithms.

10.3 Directions for Future Research

There are a number of research directions that stem from the work described in this thesis. These future areas of research are recommended below

1. The order of general class function dynamics has a direct relationship with the number of d.o.f. or n_c. In most of the cases, the higher order function parameterisation improves the trade off within OMPC as it was observed from the simulation results (in Chapter 5). An interesting future direction is to formulate H_G using higher order function parameterisation without increasing n_c.

2. In Chapter 6, there is a limitation on the comparison of closed loop performance due to the infeasibility of different algorithms, in the author's view further work need to be done to propose a robust scenario to overcome this and thus improve the multiobjective optimisation.

3. The proposed multiobjective optimisation can include further objectives e.g. order of parameterisation dynamics, number of inequalities etc.

4. Another interesting future direction is to optimise the parameterised matrix H_G directly using multiobjective optimisation.

5. The proposed multi-objective optimisation can be simplified to single objective optimisation using a prior specification of β and γ.

6. There is a need to consider the associated quadratic programming problems in more detail and in particular to consider to what extent the general class of parameterisations either restrict or enable highly structured optimisations which thus are amenable to efficient coding; it is known that OMPC does have a good structure that can be exploited.

7. This thesis considers parameter uncertainty only for robustness in chapter 8 and 9. Some interesting future research directions are the extension of the algorithms to systems subject to bounded disturbances using augmented dynamics.

8. In general, introducing robustness within the MPC problem formulation is computationally too demanding for practical implementation. It will be an interesting future direction to consider the computational efficiency for multiparametric quadratic programming (mp-QP) solutions to propose algorithms using a general class of function parameterisation. This can further extend to a systematic selection of generalised function parameterisation based on the region of attraction and the resulting number of regions for parametric solution.

9. It is strongly recommended to test all the proposed algorithms in standard hardware.

Appendix A

State space form of generalised functions

This appendix presents the state space form of Kautz and generalised function.

A.1 State space form of Kautz function

The Kautz network is defined as follows

$$k_i(z) = \sqrt{(1-a^2)(1-b^2)}\frac{(z^{-1}-a)^{i-1}(z^{-1}-b)^{i-1}}{(1-az^{-1})^i(1-bz^{-1})^i}; \qquad (A.1)$$
$$0 \leq a < 1; \;\; 0 \leq b < 1$$

where 'a' and 'b' are poles of the discrete-time Kautz network. However, the inverse z-transform of the Kautz networks do not lead to a compact expression of the Kautz functions in the time-domain so state-space representation is preferred and derived briefly.

The z-transforms of the discrete-time Kautz functions are written as

$$\begin{aligned} k_0(z) &= \frac{\sqrt{(1-a^2)(1-b^2)}}{1-az^{-1}} \\ k_1(z) &= \frac{1}{(1-bz^{-1})}k_0(z) \\ k_2(z) &= \frac{z^{-1}-a}{(1-az^{-1})}k_1(z) \\ &\vdots \\ k_N(z) &= \frac{(z^{-1}-b)}{(1-bz^{-1})}k_{N-1}(z) \end{aligned} \tag{A.2}$$

where $0 \leq (a, b) < 1$ for stability of the functions.

The discrete time network (A.2) can be expressed in difference equation as

$$\begin{aligned} k_0(n) &= ak_0(n-1) + \sqrt{(1-a^2)(1-b^2)} \\ k_1(n) &= bk_1(n-1) + \sqrt{(1-a^2)(1-b^2)} \\ k_2(n) &= ak_2(n-1) + (1-ab)k_1(n-1) - a\sqrt{(1-a^2)(1-b^2)} \\ k_3(n) &= bk_3(n-1) + (1-ab)k_2(n-1) - b(1-ab)k_1(n-1) + ab\sqrt{(1-a^2)(1-b^2)} \\ k_4(n) &= ak_4(n-1) + (1-ab)k_3(n-1) - a(1-ab)k_2(n-1) + ab(1-ab)k_1(n-1) \\ &\quad - a^2b\sqrt{(1-a^2)(1-b^2)} \\ k_5(n) &= bk_5(n-1) + (1-ab)k_4(n-1) - b(1-ab)k_3(n-1) + ab(1-ab)k_2(n-1) \\ &\quad - ab^2(1-ab)k_1(n-1) + a^2b^2\sqrt{(1-a^2)(1-b^2)} \\ k_6(n) &= ak_6(n-1) + (1-ab)k_5(n-1) - b(1-ab)k_4(n-1) + ab(1-ab)k_3(n-1) \\ &\quad - ab^2(1-ab)k_2(n-1) + a^2b^2(1-ab)k_1(n-1) - a^3b^2\sqrt{(1-a^2)(1-b^2)} \\ &\vdots \end{aligned} \tag{A.3}$$

In state space form

$$\mathcal{K}_n = \begin{bmatrix} k_1(n) \\ k_2(n) \\ k_3(n) \\ k_4(n) \\ k_5(n) \\ k_6(n) \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} b & 0 & 0 & 0 & 0 & 0 \\ (1-ab) & a & 0 & 0 & 0 & 0 \\ -b(1-ab) & (1-ab) & b & 0 & 0 & 0 \\ ab(1-ab) & -b(1-ab) & (1-ab) & a & 0 & 0 \\ -ab^2(1-ab) & ab(1-ab) & -b(1-ab) & (1-ab) & b & 0 \\ a^2b^2(1-ab) & -ab^2(1-ab) & ab(1-ab) & -b(1-ab) & (1-ab) & a \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{A_K} \mathcal{K}_{n-1}$$

$$\mathcal{K}_0 = \sqrt{(1-a^2)(1-b^2)} \begin{bmatrix} 1 \\ -a \\ ab \\ -a^2b \\ a^2b^2 \\ -a^3b^2 \\ \vdots \end{bmatrix} \tag{A.4}$$

A.2 State space form of generalised function

The generalised network for example, in case of $4th$ order orthonormal basis function is given by

$$g_i(z) = \sqrt{(1-a_i^2)\dots(1-a_4^2)}\frac{(z^{-1}-a_1)^{i-1}\dots(z^{-1}-a_4)^{i-1}}{(1-a_1z^{-1})^i\dots(1-a_4z^{-1})^i} \tag{A.5}$$
$$0 \le a_j < 1, \qquad j = 1, \dots, 4,$$

The z-transforms of the discrete-time Kautz functions are written as

$$
\begin{aligned}
g_0(z) &= \frac{\sqrt{(1-a_1^2)\dots(1-a_4^2)}}{1-a_1z^{-1}} \\
g_1(z) &= \frac{1}{(1-a_2z^{-1})}g_0(z) \\
g_2(z) &= \frac{1}{(1-a_3z^{-1})}g_1(z) \\
g_3(z) &= \frac{1}{(1-a_4z^{-1})}g_2(z) \\
g_4(z) &= \frac{z^{-1}-a_1}{(1-a_1z^{-1})}g_3(z) \\
g_5(z) &= \frac{z^{-1}-a_2}{(1-a_2z^{-1})}g_4(z) \\
g_6(z) &= \frac{z^{-1}-a_3}{(1-a_3z^{-1})}g_5(z) \\
&\vdots \\
g_N(z) &= \frac{(z^{-1}-a_1)}{(1-a_1z^{-1})}g_{N-1}(z)
\end{aligned}
\tag{A.6}
$$

The discrete time network (A.6) can be expressed in difference equation as

$$
\begin{aligned}
g_0(n) &= \sqrt{(1-a_1^2)\dots(1-a_4^2)} \\
g_1(n) &= a_2g_1(n-1) + \sqrt{(1-a_1^2)\dots(1-a_4^2)} \\
g_2(n) &= a_3g_2(n-1) + a_2g_1(n-1) + \sqrt{(1-a_1^2)\dots(1-a_4^2)} \\
g_3(n) &= a_4g_3(n-1) + a_3g_2(n-1) + a_2g_1(n-1) + \sqrt{(1-a_1^2)\dots(1-a_4^2)} \\
g_4(n) &= a_1g_4(n-1) + (1-a_1a_4)g_3(n-1) - a_1a_3g_2(n-1) - a_1a_2g_1(n-1) \\
&\quad - a_1\sqrt{(1-a_1^2)\dots(1-a_4^2)} \\
g_5(n) &= a_2g_5(n-1) + (1-a_1a_2)g_4(n-1) - a_2(1-a_1a_4)g_3(n-1) + a_1a_2a_3g_2(n-1) \\
&\quad + ab^2g_1(n-1) + a_1a_2\sqrt{(1-a_1^2)\dots(1-a_4^2)} \\
g_6(n) &= a_3g_6(n-1) + (1-a_2a_3)g_5(n-1) - a_3(1-a_1a_2)g_4(n-1) + a_2a_3(1-a_1a_4)g_3(n-1) \\
&\quad - a_1a_2a_3^2g_2(n-1) - a_1a_2^2a_3g_1(n-1) - a_1a_2a_3\sqrt{(1-a_1^2)\dots(1-a_4^2)} \\
&\vdots
\end{aligned}
\tag{A.7}
$$

In state space form

$$G_n = \begin{bmatrix} g_1(n) \\ g_2(n) \\ g_3(n) \\ g_4(n) \\ g_5(n) \\ g_6(n) \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} a_2 & 0 & 0 & 0 & 0 & 0 \\ a_2 & a_3 & 0 & 0 & 0 & 0 \\ a_2 & a_3 & a_4 & 0 & 0 & 0 \\ -a_1a_2 & -a_1a_3 & (1-a_1a_4) & a_1 & 0 & 0 \\ a_1a_2^2 & a_1a_2a_3 & -a_2(1-a_1a_4) & (1-a_1a_2) & a_2 & 0 \\ -a_1a_2^2a_3 & -a_1a_2a_3^2 & a_2a_3(1-a_1a_4) & -a_3(1-a_1a_2) & (1-a_1a_2) & a_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_{A_G} G_{n-1}$$

$$G_0 = \sqrt{(1-a_1^2)\dots(1-a_4^2)} \begin{bmatrix} 1 \\ 1 \\ 1 \\ -a_1 \\ a_1a_2 \\ -a_1a_2a_3 \\ \vdots \end{bmatrix} \tag{A.8}$$

Bibliography

[1] R. Murray, *Control in an information rich world: Report of the panel on future directions in control, dynamics, and systems.* Society for Industrial Mathematics, 1987.

[2] L. Wang, *Model predictive control system design and implementation using MATLAB*, London, 2009.

[3] E. Camacho and C. Bordons, *Model predictive control.* London: Springer, 2004.

[4] J. Maciejowski, *Predictive control with constraints.* London: Prentice Hall, 2002.

[5] J. Rossiter, *Model-based predictive control, a practical approach.* London: CRC Press, 2003.

[6] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.

[7] T. F. Edgar, "Control and operations: when does controllability equal profitability?" *Computers & chemical engineering*, vol. 29, no. 1, pp. 41–49, 2004.

[8] P. Dufour, "Control engineering in drying technology: review and trends," *Drying technology*, vol. 24, no. 7, pp. 889–904, 2006.

[9] S. Qin and T. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.

[10] J. Rossiter, B. Kouvaritakis, and M. J. Rice, "A numerically robust state-space approach to stable predictive control," *Automatica*, vol. 34, pp. 65–73, 1998.

[11] P. Scokaert and J. Rawlings, "Constrained linear quadratic regulation," *IEEE Trans. on Aut. Control*, vol. 43, pp. 1163–1169, 1998.

[12] E. Gilbert and K. Tan, "Linear systems with state and control constraints: the theory and application of maximal output admissible sets," *Trans. IEEE AC*, vol. 36, pp. 1008–1020, 1991.

[13] B. Kouvaritakis, J. Rossiter, and M. Cannon, "Linear quadratic feasible predictive control," *Automatica*, vol. 34, pp. 1583–1592, 1998.

[14] J. Mendez, B. Kouvaritakis, and J. Rossiter, "State-space approach to interpolation in MPC," *International Journal of Control*, vol. 10, pp. 27–38, 2000.

[15] J. Rossiter, B. Kouvaritakis, and M. Bacic, "Interpolation based computataionally efficient predictive control," *International Journal of Control*, vol. 77, pp. 290–301, 2004.

[16] J. Rossiter and P. Grieder, "Using interpolation to improve efficiency of multiparametric predictive control," *Automatica*, vol. 41, no. 4, pp. 637–643, 2005.

[17] R. Cagienard, P. Grieder, E.C.Kerrigan, and M. Morari, "Move blocking strategies in receding horizon," *Journal of Process Control*, vol. 17, pp. 563–570, 2007.

[18] L. Wang, "Discrete model predictive controller design using Laguerre functions," *Journal of Process Control*, vol. 14, pp. 131–142, 2004.

[19] E. van Donkelaar, O. Bosgra, and P. V. den Hof Khan, "Model predictive control with generalized input parameterization," *Proc. European Control Conference*, 1999.

[20] R. Gondalaker, J. Imura, and K. Kashima, "Controller invariant feasibility - a general approach to forcing strong feasibility in mpc applied to move blocking," *Automatica*, vol. 45, pp. 2869–2875, 2009.

[21] B. Kouvaritakis, J. Rossiter, and J. Schuurmans, "Efficient robust predictive control," *Automatic Control, IEEE Transactions on*, vol. 45, no. 8, pp. 1545–1549, 2000.

[22] B. Kouvaritakis, M. Cannon, and J. Rossiter, "Who needs qp for linear mpc anyway?" *Automatica*, vol. 38, no. 5, pp. 879–884, 2002.

[23] S. Drageset, L. Imsland, and B. Foss, "Efficient model predictive control with prediction dynamics," in *Proc. 7th European Control Conference ECC03*. Citeseer, 2003.

[24] L. Imsland, N. Bar, and B. Foss, "More efficient predictive control," *Automatica*, vol. 41, no. 8, pp. 1395–1403, 2005.

[25] M. Cannon and B. Kouvaritakis, "Optimizing prediction dynamics for robust MPC," *Automatic Control, IEEE Transactions on*, vol. 50, no. 11, pp. 1892–1897, 2005.

[26] J. Rossiter, L. Wang, and G. Valencia-Palomo, "Efficient algorithms for trading off feasibility and performance in predictive control," *International Journal of Control*, vol. 83, pp. 789–797, 2010.

[27] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control: Applications to industrial processes," *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[28] C. Garcia, D. Prett, and M. Morari, "Model predictive control: theory and practicea survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[29] J. Lee and B. Cooley, "Recent advances in model predictive control and other related areas," in *AIChE Symposium Series*, vol. 93, no. 316. New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997, pp. 201–216.

[30] A. Bemporad and M. Morari, "Robust model predictive control: A survey," *Robustness in identification and control*, pp. 207–226, 1999.

[31] J. Rawlings, "Tutorial overview of model predictive control," *Control Systems, IEEE*, vol. 20, no. 3, pp. 38–52, 2000.

[32] G. Goodwin, M. Seron, and J. De Doná, *Constrained control and estimation: an optimisation approach*. Springer, 2004.

[33] W. Kwon and S. Han, *Receding horizon control: model predictive control for state models*, London, 2005.

[34] B. Kouvaritakis and M. Cannon, "Non-linear predictive control: theory and practice." Institution of Electrical Engineers, 2001.

[35] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*. Birkhäuser, 2000, vol. 26.

[36] N. Ricker, "Model predictive control: State of the art," in *Chemical process controlCPC IV, Fourth international conference on chemical process control*, 1991, pp. 271–296.

[37] J. Richalet, "Industrial applications of model based predictive control," *Automatica*, vol. 29, no. 5, pp. 1251–1274, 1993.

[38] M. Morari and J. H Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4, pp. 667–682, 1999.

[39] F. Allgöwer, T. Badgwell, J. Qin, J. Rawlings, and S. Wright, "Nonlinear predictive control and moving horizon estimation–an introductory overview," *Advances in Control, Highlights of ECC*, vol. 99, pp. 391–449, 1999.

[40] R. Findeisen, L. Imsland, F. Allgower, and B. Foss, "State and output feedback nonlinear model predictive control: An overview," *European journal of control*, vol. 9, no. 2-3, pp. 190–206, 2003.

[41] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," *Nonlinear model predictive control*, pp. 345–369, 2009.

[42] M. Morari and N. Ricker, *Model Predictive Control Toolbox for Use with MATLAB®: User's Guide*. MathWorks, Incorporated, 1998.

[43] A. Bemporad, M. Morari, and N. Ricker, "Model predictive control toolbox," *User's Guide, Version*, vol. 2, 2004.

[44] E. Lee and L. Markus, "Foundations of optimal control theory," DTIC Document, Tech. Rep., 1967.

[45] W. H. Fleming and R. W. Rishel, *Deterministic and stochastic optimal control*. New York Springer, 1975.

[46] R. Kalman *et al.*, "Contributions to the theory of optimal control," *Bol. Soc. Mat. Mex*, vol. 5, no. 1, pp. 102–119, 1960.

[47] ——, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[48] R. Bellman, "On the theory of dynamic programming," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 38, no. 8, p. 716, 1952.

[49] B. Gidas, W.-M. Ni, and L. Nirenberg, "Symmetry and related properties via the maximum principle," *Communications in Mathematical Physics*, vol. 68, no. 3, pp. 209–243, 1979.

[50] L. Zadeh and B. Whalen, "On optimal control and linear programming," *Automatic Control, IRE Transactions on*, vol. 7, no. 4, pp. 45–46, 1962.

[51] A. Propoi, "Use of linear programming methods for synthesizing sampled-data automatic systems," *Automation and Remote Control*, vol. 24, no. 7, pp. 837–844, 1963.

[52] T. Chang and D. Seborg, "A linear programming approach for multivariable feedback control with inequality constraints," *International journal of control*, vol. 37, no. 3, pp. 583–597, 1983.

[53] P. Campo and M. Morari, "Model predictive optimal averaging level control," *AIChE journal*, vol. 35, no. 4, pp. 579–591, 1989.

[54] C. V. Rao and J. B. Rawlings, "Linear programming and model predictive control," *Journal of Process Control*, vol. 10, no. 2, pp. 283–289, 2000.

[55] H. Genceli and M. Nikolaou, "Robust stability analysis of constrained l1-norm model predictive control," *AIChE Journal*, vol. 39, no. 12, pp. 1954–1965, 2004.

[56] D. Kleinman, "An easy way to stabilize a linear constant system," *Automatic Control, IEEE Transactions on*, vol. 15, no. 6, pp. 692–692, 1970.

[57] Y. Thomas, "Linear quadratic optimal estimation and control with receding horizon," *Electronics Letters*, vol. 11, no. 1, pp. 19–21, 1975.

[58] W. Kwon and A. Pearson, "A modified quadratic cost problem and feedback stabilization of a linear system," *Automatic Control, IEEE Transactions on*, vol. 22, no. 5, pp. 838–842, 1977.

[59] W. Kwon, A. Bruckstein, and T. Kailath, "Stabilizing state-feedback design via the moving horizon method," *International Journal of Control*, vol. 37, no. 3, pp. 631–643, 1983.

[60] J. Richalet, A. Rault, J. Testud, and J. Papon, "Algorithmic control of industrial processes," in *Proceedings of the 4th IFAC symposium on identification and system parameter estimation*. URSS September, Tbilisi,, 1976, pp. 1119–1167.

[61] C. Cutlar and B. Ramarker, "Dynamic matrix control – a computer control algorithm," *Proc. 86th National meeting of the AIChE, Houston, TX*, 1979.

[62] C. Cutler and B. Ramaker, "Dynamic matrix control–a computer control algorithm," in *Proc. Joint Automatic Control Conference, San Francisco, CA.*, 1980.

[63] D. Prett and R. Gillette, "Optimization and constrained multivariable control of a catalytic cracking unit," in *Proceedings of the joint automatic control conference*, vol. 1, 1980.

[64] E. Carlos and A. Morshedi, "Quadratic programming solution of dynamic matrix control (qdmc)," *Chemical Engineering Communications*, vol. 46, no. 1-3, pp. 73–87, 1986.

[65] P. Marquis and J. Broustail, "Smoc, a bridge between state space and model predictive controllers: application to the automation of a hydrotreating unit," in *Proceedings of the IFAC workshop on model based process control*, 1988, pp. 37–43.

[66] V. Peterka, "Predictor-based self-tuning control," *Automatica*, vol. 20, no. 1, pp. 39–50, 1984.

[67] B. Ydstie, "Extended horizon adaptive control," in *Proceedings of the 9th IFAC World Congress*, vol. 7, 1984, pp. 133–138.

[68] C. Greco, G. Menga, E. Mosca, and G. Zappa, "Performance improvements of self-tuning controllers by multistep horizons: The musmar approach," *Automatica*, vol. 20, no. 5, pp. 681–699, 1984.

[69] R. De Keyser and A. Van Cauwenberghe, "Extended prediction self-adaptive control," in *IFAC Symp. on Identification and System Parameter Estimation*, 1985, pp. 1255–1260.

[70] S. Keerthi and E. G. Gilbert, "Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations," *Journal of optimization theory and applications*, vol. 57, no. 2, pp. 265–293, 1988.

[71] D. Clarke, C. Mohtadi, and P. Tuffs, "Generalised predictive control, part 1. the basic algorithm," *Automatica*, vol. 23, pp. 137–148, 1987a.

[72] ——, "Generalised predictive control, part 2. extension and interpretations," *Automatica*, vol. 23, pp. 149–163, 1987b.

[73] D. Clarke and P. Gawthrop, "Self-tuning control," *Electrical Engineers, Proceedings of the Institution of*, vol. 126, no. 6, pp. 633–640, 1979.

[74] H. Michalska and D. Mayne, "Robust receding horizon control of constrained nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 38, no. 11, pp. 1623–1633, 1993.

[75] W. Kwon, "Advances in predictive control: Theory and application," in *Asian Control Conference, Tokyo*, 1994.

[76] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux Meeting on Systems and Control*, vol. 11, 2002.

[77] G. De Nicolao, L. Magni, and R. Scattolini, "On the robustness of receding-horizon control with terminal constraints," *Automatic Control, IEEE Transactions on*, vol. 41, no. 3, pp. 451–453, 1996.

[78] L. Magni and R. Sepulchre, "Stability margins of nonlinear receding horizon control via inverse optimity," *System & Control Letters*, vol. 32, pp. 241–245, 1997.

[79] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[80] P. Scokaert and D. Mayne, "Min-max feedback model predictive control for constrained linear systems," *Automatic Control, IEEE Transactions on*, vol. 43, no. 8, pp. 1136–1142, 1998.

[81] Y. Lee and B. Kouvaritakis, "Robust receding horizon predictive control for systems with uncertain dynamics and input saturation," *Automatica*, vol. 36, no. 10, pp. 1497–1504, 2000.

[82] D. Mayne, M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[83] B. Pluymers, J. Rossiter, J. Suykens, and B. De Moor, "A simple algorithm for robust MPC," in *Proceedings of the IFAC World Congress*, 2005.

[84] P. Campo and M. Morari, "Robust model predictive control," in *American Control Conference, 1987*. IEEE, 1987, pp. 1021–1026.

[85] A. Casavola, M. Giannelli, and E. Mosca, "Min–max predictive control strategies for input-saturated polytopic uncertain systems," *Automatica*, vol. 36, no. 1, pp. 125–133, 2000.

[86] B. Ding, Y. Xi, and S. Li, "A synthesis approach of on-line constrained robust model predictive control," *Automatica*, vol. 40, no. 1, pp. 163–167, 2004.

[87] E. Kerrigan and J. Maciejowski, "Feedback min-max model predictive control using a single linear program: Robust stability and the explicit solution," *Int. J. Robust Nonlinear Control*, vol. 14, pp. 395–413, 2004.

[88] J. Allwright, "On min-max model-based predictive control," 1993.

[89] J. Allwright and G. Papavasiliou, "On linear programming and robust model-predictive control using impulse-responses," *Systems & Control Letters*, vol. 18, no. 2, pp. 159–164, 1992.

[90] J. Lee and Z. Yu, "Worst-case formulations of model predictive control for systems with bounded parameters," *Automatica*, vol. 33, no. 5, pp. 763–781, 1997.

[91] A. Bemporad, F. Borrelli, and M. Morari, "Min-max control of constrained uncertain discrete-time linear systems," *Automatic Control, IEEE Transactions on*, vol. 48, no. 9, pp. 1600–1606, 2003.

[92] E. Zafiriou, "Robust model predictive control of processes with hard constraints," *Computers & Chemical Engineering*, vol. 14, no. 4, pp. 359–371, 1990.

[93] A. Zheng and M. Morari, "Robust control of linear time-varying systems with constraints," pp. 205–220, 1995.

[94] T. Yang and E. Polak, "Moving horizon control of nonlinear systems with input saturation, disturbances and plant uncertainty," *International Journal of Control*, vol. 58, no. 4, pp. 875–903, 1993.

[95] S. L. de Oliveira Kothare and M. Morari, "Contractive model predictive control for constrained nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 45, no. 6, pp. 1053–1071, 2000.

[96] H. Genceli and M. Nikolaou, "Robust stability analysis of constrained l1-norm model predictive control," *AIChE Journal*, vol. 39, no. 12, pp. 1954–1965, 1993.

[97] J. H. Lee and B. L. Cooley, "Stable min-max control for state-space systems with bounded input matrix," in *American Control Conference, 1997. Proceedings of the 1997*, vol. 5. IEEE, 1997, pp. 2945–2949.

[98] T. A. Badgwell, "Robust model predictive control of stable linear systems," *International Journal of Control*, vol. 68, no. 4, pp. 797–818, 1997.

[99] A. Bemporad, "Reducing conservativeness in predictive control of constrained systems with disturbances," in *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, vol. 2. IEEE, 1998, pp. 1384–1389.

[100] A. Brooms, B. Kouvaritakis, and Y. Lee, "Constrained mpc for uncertain linear systems with ellipsoidal target sets," *Systems & Control Letters*, vol. 44, no. 3, pp. 157–166, 2001.

[101] C. Løvaas, M. M. Seron, and G. C. Goodwin, "Robust output-feedback model predictive control for systems with unstructured uncertainty," *Automatica*, vol. 44, no. 8, pp. 1933–1943, 2008.

[102] X. Li and T. E. Marlin, "Model predictive control with robust feasibility," *Journal of Process Control*, vol. 21, no. 3, pp. 415–435, 2011.

[103] Z. Wan and M. V. Kothare, "An efficient off-line formulation of robust model predictive control using linear matrix inequalities," *Automatica*, vol. 39, no. 5, pp. 837–846, 2003.

[104] ——, "Efficient robust constrained model predictive control with a time varying terminal constraint set," *Systems & control letters*, vol. 48, no. 5, pp. 375–383, 2003.

[105] B. Pluymers, J. Suykens, and B. De Moor, "Min–max feedback mpc using a time-varying terminal constraint set and comments on efficient robust constrained model predictive control with a time-varying terminal constraint set?" *Systems & control letters*, vol. 54, no. 12, pp. 1143–1148, 2005.

[106] Z. Wan, B. Pluymers, M. Kothare, and B. De Moor, "Comments on:efficient robust constrained model predictive control with a time varying terminal constraint set by wan and kothare," *Systems & Control Letters*, vol. 55, no. 7, pp. 618–621, 2006.

[107] D. Angeli, A. Casavola, and E. Mosca, "Ellipsoidal low-demanding mpc schemes for uncertain polytopic discrete-time systems," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3. IEEE, 2002, pp. 2935–2940.

[108] B. Ding, Y. Xi, M. T. Cychowski, and T. OMahony, "Improving off-line approach to robust mpc based-on nominal performance cost," *Automatica*, vol. 43, no. 1, pp. 158–163, 2007.

[109] Y. I. Lee and B. Kouvaritakis, "Constrained robust model predictive control based on periodic invariance," *Automatica*, vol. 42, no. 12, pp. 2175–2181, 2006.

[110] D.-W. LI and Y.-G. XI, "Design of robust model predictive control based on multi-step control set," *Acta Automatica Sinica*, vol. 35, no. 4, pp. 433–437, 2009.

[111] D. Li and Y. Xi, "The feedback robust mpc for lpv systems with bounded rates of parameter changes," *Automatic Control, IEEE Transactions on*, vol. 55, no. 2, pp. 503–507, 2010.

[112] J. Rossiter, B. Kouvaritakis, and M. Cannon, "Triple mode control in MPC," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 6. IEEE, 2000, pp. 3753–3757.

[113] ——, "Computationally efficient algorithms for constraint handling with guaranteed stability and near optimality," *International Journal of Control*, vol. 74, no. 17, pp. 1678–1689, 2001.

[114] L. Imsland, J. Rossiter, B. Pluymers, and J. Suykens, "Robust triple mode MPC," *International Journal of Control*, vol. 81, pp. 679–689, 2008.

[115] B. Pluymers, J. Rossiter, J. Suykens, and B. De Moor, "The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty," in

American Control Conference, 2005. Proceedings of the 2005. IEEE, 2005, pp. 804–809.

[116] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit solution of model predictive control via multiparametric quadratic programming," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 2. IEEE, 2000, pp. 872–876.

[117] A. Bemporad and C. Filippi, "Suboptimal explicit mpc via approximate multiparametric quadratic programming," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 5. IEEE, 2001, pp. 4851–4856.

[118] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[119] P. Tøndel, T. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit mpc solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.

[120] A. Grancharova and T. Johansen, "Explicit approaches to constrained model predictive control: a survey," *Modeling, identification and control*, vol. 25, no. 3, pp. 131–157, 2004.

[121] ——, "Survey of explicit approaches to constrained optimal control," *Switching and Learning in Feedback Systems*, pp. 47–97, 2005.

[122] A. Bemporad and C. Filippi, "An algorithm for approximate multiparametric convex programming," *Computational optimization and applications*, vol. 35, no. 1, pp. 87–108, 2006.

[123] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari, "Multi-parametric toolbox (mpt)," *Hybrid systems: computation and control*, pp. 121–124, 2004.

[124] T. Johansen, T. Fossen, and P. Tøndel, "Efficient optimal constrained control allocation via multiparametric programming," *Journal of guidance, control, and dynamics*, vol. 28, no. 3, pp. 506–515, 2005.

[125] E. Pistikopoulos, "Perspectives in multiparametric programming and explicit model predictive control," *AIChE Journal*, vol. 55, no. 8, pp. 1918–1925, 2009.

[126] M. Canale, L. Fagiano, and M. Milanese, "Set membership approximation theory for fast implementation of model predictive control laws," *Automatica*, vol. 45, no. 1, pp. 45–54, 2009.

[127] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. on Aut. Control*, vol. 18, pp. 279–293, 2010.

[128] L. Borrelli, T. Baotic, A. Bemporad, and T. Morari, "Efficient on-line computation of constrained optimal control," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 2. IEEE, 2001, pp. 1187–1192.

[129] P. Tøndel, T. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, 2003.

[130] F. Christophersen, M. Kvasnica, C. Jones, and M. Morari, "Efficient evaluation of piecewise control laws defined over a large number of polyhedra," in *European control conference*, 2007, pp. 2360–2367.

[131] T. Geyer, F. Torrisi, and M. Morari, "Optimal complexity reduction of piecewise affine models based on hyperplane arrangements," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2. IEEE, 2004, pp. 1190–1195.

[132] T. Johansen, I. Petersen, and O. Slupphaug, "Explicit sub-optimal linear quadratic regulation with state and input constraints," *Automatica*, vol. 38, no. 7, pp. 1099–1111, 2002.

[133] T. Johansen and A. Grancharova, "Approximate explicit constrained linear model predictive control via orthogonal search tree," *Automatic Control, IEEE Transactions on*, vol. 48, no. 5, pp. 810–815, 2003.

[134] A. Bemporad and C. Filippi, "Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming," *Journal of optimization theory and applications*, vol. 117, no. 1, pp. 9–38, 2003.

[135] M. Canale, L. Fagiano, M. Milanese, and C. Novara, "Set membership approximations of predictive control laws: the tradeoff between accuracy and complexity," *Control Theory & Applications, IET*, vol. 4, no. 12, pp. 2907–2920, 2010.

[136] G. Valencia-Palomo and J. Rossiter, "Efficient suboptimal parametric solutions to predictive control for plc applications," *Control Engineering Practice*, vol. 19, no. 7, pp. 732–743, 2011.

[137] J. Rossiter, B. Kouvaritakis, and M. Cannon, "An algorithm for reducing complexity in parametric predictive control," *International Journal of Control*, vol. 78, pp. 1511–1520, 2005.

[138] G. Valencia-Palomo and J. Rossiter, "Using laguerre functions to improve efficiency of multi-parametric predictive control," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 4731–4736.

[139] R. Fletcher, *Practical Methods of Optimization*. Wiley-Interscience New York, NY, USA, 1987.

[140] M. Hintermüller, K. Ito, and K. Kunisch, "The primal-dual active set strategy as a semismooth newton method," *SIAM Journal on Optimization*, vol. 13, no. 3, pp. 865–888, 2002.

[141] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit mpc," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.

[142] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[143] S. Wright, *Primal-dual interior-point methods*. Society for Industrial Mathematics, 1987, vol. 54.

[144] C. Rao, S. Wright, and J. Rawlings, "Application of interior-point methods to model predictive control," *Journal of optimization theory and applications*, vol. 99, no. 3, pp. 723–757, 1998.

[145] A. Hansson, "A primal-dual interior-point method for robust optimal control of linear discrete-time systems," *Automatic Control, IEEE Transactions on*, vol. 45, no. 9, pp. 1639–1655, 2000.

[146] M. Todd, "The many facets of linear programming," *Mathematical Programming*, vol. 91, no. 3, pp. 417–436, 2002.

[147] V. Klee and G. J. Minty, "How good is the simplex method," *Inequalities-III*, pp. 159–175, 1972.

[148] K. H. Borgwardt, *A probabilistic analysis of the simplex method*. Springer-Verlag New York, Inc., 1986.

[149] M. Kozlov, S. P. Tarasov, and L. Khachiyan, "The polynomial solvability of convex quadratic programming," *USSR Computational Mathematics and Mathematical Physics*, vol. 20, no. 5, pp. 223–228, 1980.

[150] A. Schrijver, *Combinatorial optimization: polyhedra and efficiency*. Springer Verlag, 2003, vol. 24.

[151] C. Maes, "A regularized active-set method for sparse convex quadratic programming," Ph.D. dissertation, Stanford University, 2010.

[152] R. Bartlett, A. Wachter, and L. Biegler, "Active set vs. interior point strategies for model predictive control," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 6. IEEE, 2000, pp. 4229–4233.

[153] P. Gill, W. Murray, and M. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM journal on optimization*, vol. 12, no. 4, pp. 979–1006, 2002.

[154] H. Ferreau, P. Ortner, P. Langthaler, L. Re, and M. Diehl, "Predictive control of a real-world diesel engine using an extended online active set strategy," *Annual Reviews in Control*, vol. 31, no. 2, pp. 293–301, 2007.

[155] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM, 1984, pp. 302–311.

[156] J. Gondzio and A. Grothey, "A new unblocking technique to warmstart interior point methods based on sensitivity analysis," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1184–1210, 2008.

[157] E. Yildirim and S. Wright, "Warm-start strategies in interior-point methods for linear programming," *SIAM Journal on Optimization*, vol. 12, no. 3, pp. 782–810, 2002.

[158] E. John and E. Yıldırım, "Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension," *Computational Optimization and Applications*, vol. 41, no. 2, pp. 151–183, 2008.

[159] Y. Wang and S. Boyd, "Fast evaluation of quadratic control-lyapunov policy," *Control Systems Technology, IEEE Transactions on*, vol. 19, no. 4, pp. 939–946, 2011.

[160] S. Richter, C. N. Jones, and M. Morari, "Real-time input-constrained mpc using fast gradient methods," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 7387–7393.

[161] A. Shahzad, E. Kerrigan, and G. Constantinides, "A fast well-conditioned interior point method for predictive control," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 508–513.

[162] ——, "A warm-start interior-point method for predictive control," *Proc. UKACC*, 2010.

[163] M. Zeilinger, C. Jones, and M. Morari, "Real-time suboptimal model predictive control using a combination of explicit mpc and online optimization," *Automatic Control, IEEE Transactions on*, vol. 56, no. 7, pp. 1524–1534, 2011.

[164] S. Richter, C. Jones, and M. Morari, "Computational complexity certification for real-time mpc with input constraints¡ newline/¿ based on the fast gradient method," *Automatic Control, IEEE Transactions on*, vol. 57, no. 6, pp. 1391–1403, 2012.

[165] M. Sznaier and M. Damborg, "Heuristically enhanced feedback control of constrained discrete-time linear systems," *Automatica*, vol. 26, no. 3, pp. 521–532, 1990.

[166] M. Bacic, M. Cannon, Y. Lee, and B. Kouvaritakis, "General interpolation in MPC and its advantages," *IEEE Transactions on Automatic Control*, vol. 48, pp. 1092–1096, 2003.

[167] J. Rossiter, B. Pluymers, and B. De Moor, "The potential of interpolation for simplifying predictive control and application to lpv systems," *Assessment and Future Directions of Nonlinear Model Predictive Control*, pp. 63–76, 2007.

[168] J. Rossiter, B. Pluymers, Y. Ding, X. Lei, J. Suykens, and B. De Moor, "Extending the feasibility of interpolation based mpc and application to lpv systems," 2005.

[169] B. Pluymers, "Robust model based predictive control-an invariant set approach," *status: published*, 2006.

[170] J. Rossiter and Y. Ding, "Interpolation methods in model predictive control: an overview," *International Journal of Control*, vol. 83, no. 2, pp. 297–312, 2010.

[171] Y. Ding and J. Rossiter, "Interpolation methods in MPC: an overview," *International Journal of Control*, vol. 83, pp. 297–312, 2010.

[172] R. Cagienard, P. Grieder, E. Kerrigan, and M. Morari, "Move blocking strategies in receding horizon control," *Journal of Process Control*, vol. 17, no. 6, pp. 563–570, 2007.

[173] R. Gondhalekar and J. Imura, "Least-restrictive move-blocking model predictive control," *Automatica*, vol. 46, no. 7, pp. 1234–1240, 2010.

[174] R. C. Shekhar and J. M. Maciejowski, "Robust variable horizon MPC with move blocking," *Systems & Control Letters*, vol. 61, no. 4, pp. 587–594, 2012.

[175] R. Gondhalekar and J.-i. Imura, "Recursive feasibility guarantees in move-blocking mpc," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 1374–1379.

[176] M. Cannon, B. Kouvaritakis, and J. Rossiter, "Efficient active set optimization in triple mode MPC," *Automatic Control, IEEE Transactions on*, vol. 46, no. 8, pp. 1307–1312, 2001.

[177] K. Tan and E. Gilbert, "Multimode controllers for linear discrete time systems with general state and control constraints," *Proc. World Scientific*, 1992.

[178] D. Naidu, *Optimal Control Systems*, ser. Electrical Engineering Textbook Series. CRC PressINC, 2003. [Online]. Available: http://books.google.co.uk/books?id=hGxurdEZVtkC

[179] K. Muske and J. Rawlings, "Model predictive control with linear models," *AIChE Journal*, vol. 39, no. 2, pp. 262–287, 1993.

[180] J. Rossiter, "A global approach to feasibility in linear mpc,," *Proc. UKACC ICC*, 2006.

[181] J. Rawlings and K. Muske, "The stability of constrained receding horizon control," *Automatic Control, IEEE Transactions on*, vol. 38, no. 10, pp. 1512–1516, 1993.

[182] D. Chmielewski and V. Manousiouthakis, "On constrained infinite-time linear quadratic optimal control," *Systems & Control Letters*, vol. 29, no. 3, pp. 121–129, 1996.

[183] H. Khalil and J. Grizzle, *Nonlinear systems*. Prentice hall New Jersey, 1996, vol. 3.

[184] M. Sznaier and M. J. Damborg, "Suboptimal control of linear systems with state and control inequality constraints," in *Decision and Control, 1987. 26th IEEE Conference on*, vol. 26. IEEE, 1987, pp. 761–762.

[185] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[186] G. Pannocchia and E. C. Kerrigan, "Offset-free receding horizon control of constrained linear systems subject to time-varying setpoints and persistent unmeasured disturbances," 2003.

[187] G. Valencia-Palomo, "Efficient implementations of predictive control," Ph.D. dissertation, University of Sheffield, 2010.

[188] L. Imsland and J. Rossiter, "Time varying terminal control," in *IFAC World Congress, Prague*, 2005.

[189] P. Goulart, E. Kerrigan, and J. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, no. 4, pp. 523–533, 2006.

[190] J. Rossiter, B. Pluymers, J. Suykens, and B. De Moor, "A multi parametric quadratic programming solution to robust predictive control," in *Proceedings of the IFAC World Congress*, 2005.

[191] S. Boyd, L. El Ghaoul, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Society for Industrial Mathematics, 1987, vol. 15.

[192] J. Schuurmans and J. Rossiter, "Robust piecewise-linear control for polytopic systems with input constraints," in *Control'98. UKACC International Conference on (Conf. Publ. No. 455)*. IET, 1998, pp. 1029–1033.

[193] M. Cannon, B. Kouvaritakis, and V. Deshmukh, "Enlargement of polytopic terminal region in NMPC by interpolation and partial invariance," *Automatica*, vol. 40, no. 2, pp. 311–317, 2004.

[194] L. Imsland, N. Bar, and B. Foss, "A new algorithm for efficient mpc and a comparison with competing schemes," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 1. IEEE, 2004, pp. 222–227.

[195] E. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. dissertation, 2000.

[196] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE, 2004, pp. 284–289.

[197] C. K. Finn, B. Wahlberg, and B. E. Ydstie, "Constrained predictive control using orthogonal expansions," *AIChE journal*, vol. 39, no. 11, pp. 1810–1826, 1993.

[198] A. Mbarek, H. Messaoud, and G. Favier, "Robust predictive control using kautz model," in *Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 184–187.

[199] X. Mingzhu, J. Yiping, P. Cunzhi, and W. Zhanzhong, "An incremental predictive functional control based on kautz model," in *Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on*. IEEE, 2009, pp. 1–5.

[200] M. Xu, Y. Jiang, J. Wen, and C. Pan, "Multiform optimization of predictive functional control based on kautz model," in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on.* IEEE, 2010, pp. 4982–4985.

[201] W. Kautz, "Transient systhensis in the time domain," *IRE Transections on Circuit Theory*, vol. 1, pp. 29–39, 1954.

[202] J. Rossiter and L. Wang, "Exploiting laguerre functions to improve the feasibility/performance compromise in MPC," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on.* IEEE, 2008, pp. 4737–4742.

[203] G. Valencia-Palomo and J. Rossiter, "Programmable logic controller implementation of an auto-tuned predictive control based on minimal plant information," *ISA Transactions*, vol. 50, pp. 92–100, 2011.

[204] R. Purshouse and P. Fleming, "On the evolutionary optimisation of many conflicting objectives," *IEEE Trans Evolutionary Computation*, vol. 11, pp. 770–784, 2007.

[205] D. A. Wolfe and M. Hollander, "Nonparametric statistical methods," *Nonparametric statistical methods*, 1973.

[206] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, vol. 18, no. 1, pp. 50–60, 1947.

[207] J. H. Lee, "Model predictive control: Review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, no. 3, pp. 415–424, 2011.

[208] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems (Lecture Notes in Control and Information Sciences). No. 290.* Springer-Verlag, 2003.

[209] P. Grieder, F. Borrel, F. Torrisi, and M. Morari, "Computation of the constrained infinite time linear quadratic regulator," *Automatica*, vol. 40, pp. 1361–1379, 2000.

[210] A. Bemporad, F. Borrelli, and M. Morari, "Optimal controllers for hybrid systems: Stability and piecewise linear explicit form," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 2. IEEE, 2000, pp. 1810–1815.

[211] J. Nocedal and S. J. Wright, *Numerical optimization.* Springer Science+ Business Media, 2006.