Multi-parametric Programming for Model Predictive Control

Thesis submitted to
National Institute of Technology, Rourkela
For award of the degree

of

Master of Technology

by

Sudipta kumar Behera
213EE3299

Under the guidance of

Prof. Asim Kumar Naskar

DEPARTMENT OF ELECTRICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

CERTIFICATE

This is to certify that the thesis entitled **Multi-parametric Programming for Model Predictive Control**, submitted by **Sudipta Kumar Behera** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirement for the awarded of the degree of Master of Technology with the specialization of **Control and Automation** in the department of **Electrical Engineering**, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Asim Kumar Naskar

(Supervisor)

____

Signature

____

Date

ACKNOWLEDGEMENTS

I express my sincere gratitude to my supervisor, Dr. Asim Kumar Naskar for his valuable guidance and suggestions without which this thesis would not be in its present form. I also thank him for his consistent encouragements throughout the work.

I express my gratitude to Prof. Anup kumar Panda, the Head of the Department of Electrical Engineering, NIT Rourkela for extending some facilities towards completion of this thesis. Thanks also to other faculty members in the department.

I thank Abhilash, Abhishek, Debashis, Anupam, Pavan and other fellow M.Tech scholars for their enjoyable and helpful company I had with.

My warmest thanks go to my family for their support, love, encouragement and patience.

Sudipta Kumar Behera

NIT Rourkela

Abstract

Model predictive control (MPC) solves a quadratic optimization problem to generate control law in each step. The usual methods of solution for quadratic optimization problem are interior point method, active set method etc. But most of the techniques are computationally heavy to perform the job in small amount of time. So a method is required where on-line computation is less. In multi-parametric quadratic programming (mp-QP) method an off-line computation is done a prior and a binary search tree is prepared. The on-line computation mainly involves a search through the binary-tree.

The mp-QP is suitable for the class of optimization problem, where the objective function is to minimize or maximize a performance criterion subject to a given set of constraints where some of the parameter vary between lower and upper bounds. Also mp-QP is suitable for multi-objective optimization, where multi criteria problems can be reformulated as multi-parametric programming problems and a parametrized optimal solution is obtained.

Multi-parametric programming is a technique for obtaining: (i) the objective and optimization variable as functions of the varying parameters and (ii) the regions in the space of the parameters where these functions are valid. The newly developed convex optimization solver CVXGEN is utilized successfully for off-line calculations which involves of dividing the parameter space into different polyhedral regions. In each one, the objective function has a constant value. The process involves another kind of optimization problem. For CVXGEN, worst case solving time is in milliseconds, even for a large problem. Thus, the use of CVXGEN minimizes the off-line calculation in mp-QP technique.

In this work, an input constraint MPC problem is chosen from existing literature. The problem is solved for both two step prediction and three step prediction cases. The parametric space is calculated using CVXGEN SDPT3 solver(a MATLAB software for semidefinite quadratic linear programming) for both the cases. The control input and states are ploted for both the MPC problems, and the results are compared.

Contents

List of Acronyms

MPC	:	Model Predictive Control
mp-QP	:	Multi-parametric Quadratic Programming
mp-NLP	:	Multi-parametric Nonlinear Programming
LQR	:	Linear Quadratic Regulator
mp-LP	:	Multi-parametric Linear Programming
mp-MIP	:	Multi-parrametric Mixed Integer Programming

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Overview

Optimization problems arises in a different engineering fields. The optimization problem involved in most of the cases in a quadratic form. The usual solution method of this problems is interior point methods, active set methods and linear programming methods. Recently multi parametric quadratic programming method is developed by A Bemporad to solve the quadratic optimization problems. This methods consists of two parts (i)off-line (ii)on-line and it is found to be usually faster than the conventional method. Multi-parametric programming is an approach for solving constrained optimization problems by computing a parameter dependent solution. It has appeared as a optimistic tool that is particularly suited for applications that need to solve optimization problems rapidly such as in model predictive control (MPC), where the value of the parameter becomes apparent on-line and the optimal control problem needs to be solved in a small fraction of the sampling period. Applications of mp programming nave also been reported for solving scheduling problems, process design and energy management in presence of uncertainties. The basic idea in the multi-parametric approach is to decompose the parameter space into separate regions, each region is define a set of optimal active constraints in the parameter space [1]. The parameter dependent solution can then be easily deduced using the necessary condition for optimality or its corresponding parametric sensitivity. Depending on the type of optimization problems, Mp-programming problems are classified as mp-linear programming, mp-quadratic programming, mp-nonlinear programming, and mp-mixed integer nonlinear programming [2].

All approaches reported in the literature for solving multi-parametric programming problems involve two basic steps: (i) determination of the optimal solution as a parameter dependent function, valid over a certain region in the parameter space and (ii) exploration of the reaming parameter space. In this thesis we developed a algorithm which is define the control action which is give the input to the process [3]. In this work, we will focus on strictly convex multi-parametric quadratic programming problems which are related to linear MPC problems with a quadratic cost function. In general the solution has the form of a piecewise affine function over a polyhedral partition of the parameter space in to so called critical regions, where each region corresponds to a set of optimal active constraints [4] [5] [6]. Parametric programming is based on the sensitivity analysis theory, distinguishing from the latter in the targets. Sensitivity analysis provides solutions in the neighbourhood of the nominal value of the varying parameters, whereas parametric programming provides a complete map of the optimal solution in the space of the varying parameters [7].

However, these widely recognized open and the closed-loop optimal control implementations involve significant on-line computations, while the control or operational action they provide only known implicitly via the solution of an optimization problem. A parametric optimization-based approach for moving off-line these rigorous calculations has been proposed in [8]; aiming to make optimization techniques applicable to a wider range of systems. The schematic description of this attractive alternative and the contrast with the traditional on-line optimization technique is shown in Fig (1.1). The key principle of this technique is that it derives off-line, before any actual process implementation occurs, the explicit mapping of the optimal decisions in the space of the plant uncertainty variations and the plant current conditions using multi-parametric programming algorithms. Thus, online optimization reduces to simple function evaluations for identifying the optimal control action. Another important advantage is that the resulting parametric control law or operational policy consists of explicit closed-form expressions that can provide precious insight into the closed-loop system features.

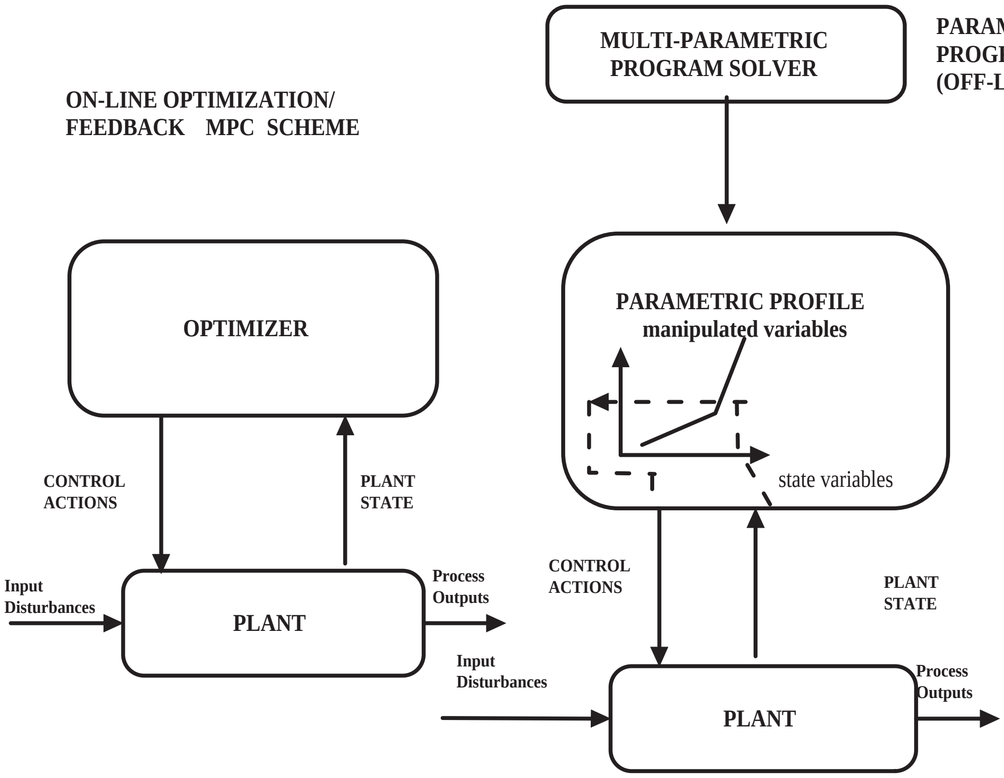

Figure 1.1: Online optimization vs. off-line parametric programming approach.

Furthermore, this novel parametric programming approach features the following advantages:

- It is not limited to steady state or discrete time dynamic systems. Thus, it portrays accurately transient plant evolution.
- It addresses directly the presence of path constraints, (e.g.,upper limits on the riser temperature in the motivating FCC example) that have to be satisfied over the complete time domain and not merely at particular time points.
- The closed-loop feedback controller derived from this technique has been developed to the extent of dealing efficiently with the presence of unpredicted or unmodeled uncertainties.
- In the presence of nonvanishing disturbances, a robust tracking controller has been designed using parametric optimization techniques.
- The explicit control law has also been designed for hybrid systems (e.g., plants that inter-mix logical discontinuous decisions with the continuous plant operation such as

the possible switch in our motivating example between the partial and the complete combustion mode).

The solution of the linear MPC optimization problem, with a quadratic objective and linear output and input constraints, by using multi-parametric programming techniques and specifically multi-parametric quadratic programming, provides a complete map of the optimal control as a function of the states and the characteristic partitions of the state space where this solution is feasible [9]. In that way the solution of the MPC problem is obtained as piecewise affine feedback control law. The on-line computational effort is small since the on-line optimization problem is solved off-line and no optimizer is ever called on-line [7]. In contrast, the on-line optimization problem is reduced to a mere function evaluation problem; when the measurements of the state are obtained and the corresponding region and control action are obtained by evaluation of a number of linear inequalities and a linear affine function, respectively. This is known as the on-line optimization via off-line parametric optimization concept.

1.2 Literature Review

A new approach for solving quadratic problems which is derived from linear MPC problem giving off-line piece-wise affine explicit solution [3] [5]. Multi-parametric programming is a term for solving an optimization problem for a range of parameter values. In multi-parametric programs, in which a vector of parameters is considered [6] [4]. Multi-parametric LP (mp-LP) is treated in [1], mp-LP in connection with MPC based on linear programming is investigated in [10]. Multi-parametric mixed-integer linear programming [1] for obtaining explicit solutions to hybrid MPC. The mp-LP algorithm [11] and mp-QP algorithm presented in this paper are similar but while [12] uses simplex steps to solve the mp-LP .

Convex optimization is widely used because it has a number of applications, e.g. control, circuit design and networking [13]. Such problems can be solved reliably and efficiently with well developed methods and tools [7], [13]. Parser solvers like CVX [9] and YALMIP [13] accepts a convex optimization problem specified in high-level language but their solve times are in the scale of seconds or minutes, which makes them unable for use in real-time systems. They also require extensive libraries and have large footprints. However in the development phase of algorithms or methods based on convex optimization, they can be a good choice as run-time and footprint are usually not great concern at any early stage (no real-time requirements).

Control Allocation is an important part of ship control systems, flight control systems and other over actuated mechanical control application [14] [15]. In this paper, demonstrated the use of the algorithm on ship control and dicussion of the control performance with the constraints control allocation. The general formulation allow several extensions compared to the mp-QP methods, since constraints limits and certain criterion parameters may be taken as parameters to the problem such that the control action may be reconfigured in the real-time. Considers how mpQP can be used for constrained control allocation in overactuated marine vessels, aircraft or other mechanical systems. In its simplest form, this is a static problem which is well suited for solution via parametric programming as the problem size is small and on-line numerical solvers are undesirable, primarily due to safety reasons [16]. The constrained control allocation problem is formulated as an mpQP and solved, giving a solution well suited for real-time implementation. Examples on over-actuated F-18 aircraft show clear improvements both in terms of on-line efficiency and optimality compared to methods from the existing literature. Experimental results for a scale model of a model ship are included. Even if I am not the first author of [17], I chose to include these results in the thesis as I contributed within formulating the problem as a parametric program and with the implementation/experiments.

1.3 Motivation

Model Predictive Control (MPC) has during the last 20 years been introduced as a highly successful control method in the process industries and chemical industries. The main reason for this success is the inherent characteristics and ability to handle constraints in complex multi-variable systems. Constraints appear in some form in most control applications and optimal performance is often obtained by operating on the constraints. In the process industries the slow processes allow real-time optimization relying on computationally demanding numerical software, while reliable low-level control takes care of fast or safety critical parts of the process. During the last few years there has been a renewed interest in multi-parametric programming within the control application. This is due to the possibility of stating constrained MPC problems as multi-parametric programs, which has allowed computationally efficient explicit solutions to problems which previously required computationally demanding real-time optimization. This thesis will treat theoretical and practical results within multi-parametric programming and its use within control applications.

1.4 Objectives

The following objective needed to specified to satisfied for better operation

- Generate the control action, which is give to the process system that should be piece-wise affine function.
- Develop an efficient algorithm to determine its parameters. The controller inherits all the stability and performance properties of model predictive control(MPC) but can be implemented without any involved on-line computations.
- Code should be simple enough to be verifiable (or at least understandable by production engineers) and also it is easy to convert to C or C+ code.
- When the code is executed that should take minimum time to execute. Worst-case execution time must be (tightly) estimated for embedding the controller in a real-time platform.Require simple/cheap hardware (microcontroller, microprocessor) and little memory to store problem data and code.
- Study the properties of the polyhedral partition of the state space where the cost function is feasible and induced by the multi-parametric piece-wise linear solution and propose a new mp-QP solver.
- Compared to existing algorithms,our approach adopts a different exploration strategy for subdividing the parameter space, avoiding unnecessary partitioning and QP proble solving.

1.5 CVXGEN

Part of this thesis is using and testing the new CVXGEN convex optimization solver which is released in 2010 by Jacob Mattingley and Stephen Boyd [13]. Testing this solver and comparing it with others is interesting because it is state-of-the-art and its applications may be used for both prototyping and real-time use.

Convex optimization is widely used because it has a number of applications, e.g. control, circuit design and networking []. Such problems can be solved reliably and efficiently with well developed methods and tools [7] [13]. Parser solvers like CVX [8] and YALMIP [7] accepts a convex optimization problem specified in high-level language but their solve times are in the scale of seconds or minutes, which makes them unable for use in real-time systems. They also require extensive libraries and have large footprints. However in the development phase of algorithms or methods based on convex optimization, they can be a good choice as run-time and footprint are usually not af great concern at any early stage (no real-time requirements).

Conventionally, the step form a general purpose parser solver to a specialized high-speed solver requires significant development time, extensive modelling and specialist knowledge of optimization and numerical algorithms. The work is also often done by hand, limiting their applications. CVXGEN is a software tool that automatically generates C-code that compiles into a convex optimization solver from a high level language specification. The C-code of the customized solvers is completely standard, standalone and extremely efficient because key structural properties of the QP problem are exploited. This leads to code with only static data structures which is almost branch-free with deterministic execution on pipeline processor architectures. The generated solvers are very reliable and robust [13] but also fast compared to parser solvers. With solve times in microseconds or milliseconds, the generated solves lend themselves to implementation in real-time applications with operation speeds in Hz or KHz. CVXGENs footprint is also simple, generating a flat, library-free solver.

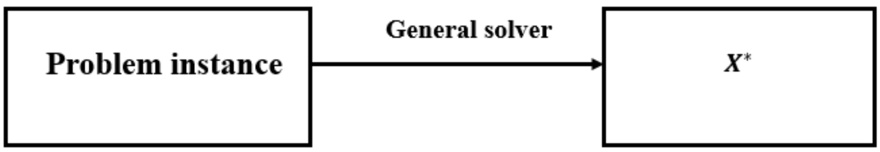

Figure 1.2: General purpose parser solver structure. Turns a single problem instance into a single optimal point.

The CVXGEN solver is currently available through a web interface on the projects web page http://www.cvxgen.com . An optimization problem specification can be entered through a MATLAB- like programming language on the web interface. Syntax specifies can be found in CVXGENs documentation [18]. The problem is entered through a fixed and structured setup, specifying problem dimensions, parameters variables, cost function and constraints.

The custom C solver is automatically generated on the web interface by the click of a button. After compilation it is available for download as a zipped archive. In addition to C code, a MATLAB interface is also available, making the custom solver available for e.g. prototyping and initial testing in the MATLAB environment. The MATLAB version will be utilized in the thesis.

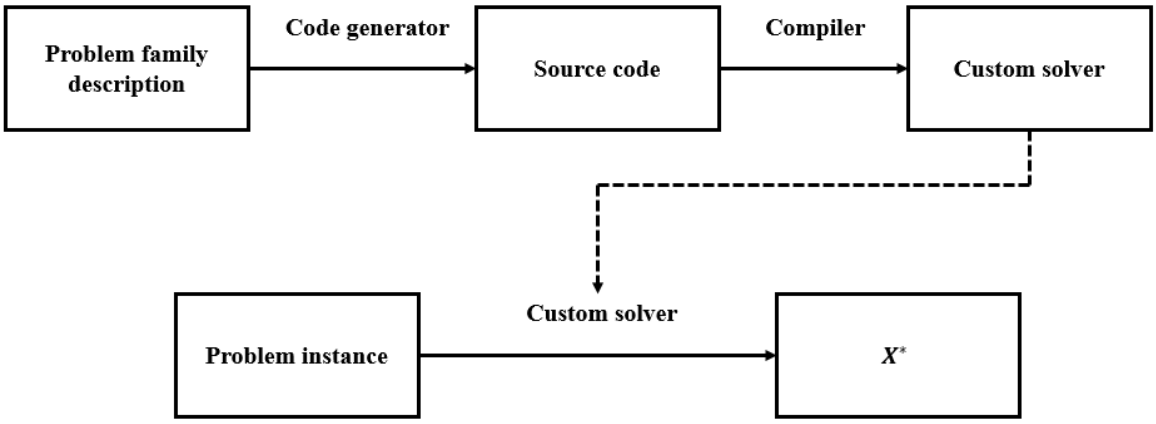

Figure 1.3: Automatic code generator solver structure. Provides optimal points for many different problem instances.

The downloaded solver is used by calling a pre-made function, with the problem instances specific parameters as function input. Solver settings can also be entered when calling the solver. After the call the solver solves the convex optimization problem with respect to the instance parameters and outputs the globally optimal variables. CVXGEN lends itself naturally to MPC problems, see [13] for a detailed overview.

1.6 Contribution and Outline

The idea of viewing an optimal control problem as a parametric program, introduced new areas of use for control schemes such as RHC. The main contributions of this thesis are within both theoretical and practical issues in the intersection between multi-parametric programming and constrained optimal control problems.

The chapter 3 is based on the papers [3] and parts of [1]. The main contribution of this thesis is the mpQP solver [3]. A strictly convex mpQP problem formulation is considered. The algorithm can be classified as an active set mpQP solver, and bears a closer resemblance to the simplex method based algorithm of (Gal 1995) than the geometric mpQP solver of [1] does. The main advantage of the method is the increased execution speed compared to other methods. Conditions are established under which the active set in a critical region can be obtained by adding or removing an element from the active set in a neighbouring critical region. The cases where these conditions are violated are handled. In particular, some results are given on how to handle degeneracies. The effect on input trajectory parametrization on explicit RHC solutions is also considered.This chapter is also based on the papers [6] [5], and considers how a PWL control law can be represented for efficient and reliable on-line implementation, by using a balanced binary search tree. The objective is to create a tree which has advantageous properties both in terms of execution time and memory requirements. An algorithm to construct such a tree is presented. It is proved that the height of such a tree is a logarithmic function of the number of regions in the PWL control law. The method has shown good results on practical problems. Moreover, a technique to obtain an approximation to a PWL control law in the form of a binary search tree is given.

The chapter 4 is a reprint of [14], which considers how mpQP can be used for constrained control allocation in overactuated marine vessels, aircraft or other mechanical systems. In its simplest form, this is a static problem which is well suited for solution via parametric programming as the problem size is small and on-line numerical solvers are undesirable, primarily due to safety reasons. The constrained control allocation problem is formulated as an mpQP and solved, giving a solution well suited for real-time implementation. Examples on over-actuated F-18 aircraft show clear improvements both in terms of on-line efficiency and optimality compared to methods from the existing literature. Experimental results for a scale model of a model ship are included. Even if I am not the first author of [17], I chose to include these results in the thesis as I contributed within formulating the problem as a parametric program and with the implementation/experiments.

CHAPTER 2

Multi-parametric Programming

2.1 Introduction

Uncertainty and variability, typically characterized by varying parameters, are inherent characteristics of any process system, it is not at all surprising then that process models, the means for translating process-related phenomena to some descriptive form (quantitative or qualitative) also involve elements of uncertainty. These varying parameters can be, for example, attributed to fluctuations in resources, technical characteristics, market requirements and prices, which can affect the feasibility and economics of a project. While the representation of the uncertainty is itself an important modelling question, the potential effect of variability on process decisions regarding process design and operations constitutes another challenging problem. Obviously the two problems are closely related: if an optimal decision is totally insensitive to the presence of uncertainty; acquiring a model for the description of the uncertainty is not really necessary. In this context, devising suitable mathematical techniques and algorithms through the application of which one could analyse and quantify if, how, what type of, and by how much, uncertainty affects decisions, becomes a major research goal.

Multi-parametric programming is a technique for solving any optimization problem, where the objective is to minimize or maximize a performance criterion subject to a given set of constraints and where some of the parameters vary between specified lower and upper bounds. The main characteristic of multi-parametric programming is its ability to obtain (i) the objective and optimization variable as functions of the varying parameters, and (ii) the regions in the space of the parameters where these functions are valid.Another important area of application of parametric programming is in multi-objective optimization, where multi-criteria problems can be reformulated as parametric programming problems and different (usually conflicting) optimal solutions, i.e., Pareto sets can be obtained as parametric solutions [2] [19].The advantage of using multi-parametric programming to address these problems is that for problems pertaining to plant operations, such as for process planning, scheduling, and control, one can obtain a complete map of all the optimal solutions. Hence, as the operating conditions vary, one does not have to re-optimize for the new set of conditions, since the optimal solution is already available as a function of the operating conditions.Depending on the type of optimization problems, Mp-programming problems are classified as four types. These are

(i)Multi-parametric Linear Programming

(ii)Multi-parametric Quadratic Programming

(iii)Multi-parametric Nonlinear Programming

(iv)Multi-parametric Mixed Integer Programming

2.2 Multi-parametric Linear Programming

When the cost function is linear and the computation of the optimal PWA function, mapping the measured state to the control input, can then be posed as the multi-parametric linear programming(MpLP).

Consider the following multiparametric linear programming(MpLP) problem

$$V^*(x) = \min_z c^T z \tag{2.1}$$

$$s.t. \quad Az = b + sx \tag{2.2}$$

$$z \geq 0 \tag{2.3}$$

where $z \in \mathbb{R}^n$ is the optimization variable, $x \in \mathbb{R}^n$ is the vector of parameters and $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $S \in \mathbb{R}^{m \times p}$ are data.If x is fixed and (2.1)-(2.2) is considered an LP, a standard way of characterizing the optimal solution is in the form of an optimal basis B. A basis is a set of indices to the z-vector, such that $z_i = 0$ for all $i \notin B$.According to the Fundamental Theorem of Linear Programming, if there exists an optimal solution to (2.1)-(2.2), at least one optimal solution is given by an optimal basis. Let N denote the non-basic variables, that is, $N = \{1, ..., q\} \setminus B$. Let A_B and A_N be the columns of A according to B and N, respectively, and z_B and z_N similarly be the corresponding elements of z. Since $z_N = 0$, we have that $A_B z_B = b + Sx$. As we have assumed that there is no degeneracy present, AB has full rank. Then,

$$z_B^*(x) = (A_B)^{-1}(b + Sx) \tag{2.4}$$

is the optimal solution whenever B is the optimal basis. Moreover, the value function is given by

$$V^*(x) = c_B^T(A_B)^{-1}(b + Sx) \tag{2.5}$$

where c_B consists of the elements corresponding to B. This means that given an optimal basis B, one can for every x such that B is an optimal basis, characterize the optimal solution $z^*$and value function V^* as linear functions of the parameter vector x.What remains is then to characterize the region in the parameter space in which B is the optimal basis. Such a region is commonly referred to as a critical region (CR). This is done by enforcing the inequality constraints (2.3). By substituting (2.4) into (2.3), one obtains

$$0 \leq (A_B)^{-1}(b + sx) \tag{2.6}$$

which is a polyhedral set in the parameter space, characterizing every $x$for which the basis B is optimal.

2.3 Multiparametric Quadratic Programming

Consider the convex quadratic mathematical program dependent on a parameter x:

$$V^*(x) = \min_z \frac{1}{2} z^T H z \tag{2.7}$$

$$s.t \quad Gz \leq W + Sx \tag{2.8}$$

where $z \in \mathbb{R}^s$ is the vector of optimization variables, $x \in \mathbb{R}^n$ is the vector of parameters, and $H \in \mathbb{R}^{s \times s}$, $G \in \mathbb{R}^{q \times s}$, $W \in \mathbb{R}^q$, and $S \in \mathbb{R}^{q \times n}$ are matrices. Here, it is supposed that $H \succ 0$, which leads to a strictly convex multi-parametric quadratic programming (mp-QP) problem (2.7)-(2.8). The case when the multi-parametric programming problem (2.7)-(2.8) is only convex, i.e. $H \succ 0$.

Let X be a polytopic set of parameters, defined by$X = \{x \in \mathbb{R}^n \,| Ax \leq b\}$. In parametric programming, it is of interest to characterize the solution of the mp-QP problem (2.7)-(2.8) for the set X.The solution of an mp-QP problem is a triple ($V^*(x)$,$Z^*(x)$,X_f),where the set of feasible parameters, $V^*\left(x\right)$ is the optimal value function, and $z^*\left(x\right)$ is the optimizer function. It is assumed that X_f is closed and $V^*\left(x\right)$ is finite for every $x \in X_f$.

An algorithm has been developed, which expresses the solution $z^*(x)$ and the optimal value $V^*(x)$ of themp-QP problem (2.7)-(2.8) as an explicit function of the parameters x, and the analytical properties of these functions have been characterized. In particular it has been proved that the solution $z^*(x)$ is a continuous piecewise linear function of x in the following sense.

Definition 1.1. A function $z(x) : X \mapsto \mathbb{R}^s$, where $X \subseteq \mathbb{R}^n$ is a polyhedral set, is piecewise linear if it is possible to partition X into convex polyhedral regions,CR_i, and $z(x) = K_i x + h_i, \forall x \in CR_i$. Piecewise quadraticity is defined analogously by letting $z(x)$ be a quadratic function $x^T Q_i x + K_i x + h_i$.

2.4 Multiparametric Nonlinear Programming

Consider the nonlinear mathematical program dependent on a parameter x appearing in the objective function and in the constraints:

$$V^*\left(x\right) = \min_{z} f(z,x) \tag{2.9}$$

$$s.t \quad g(z,x) \le 0 \tag{2.10}$$

where $z \in \mathbb{R}^n$ is the vector of optimization variables, $x \in \mathbb{R}^n$ is the vector of parameters, f is the objective function, and g is the constraints function. In (2.9), it is supposed that the minimum exists. It should be noted that the problem (2.9)-(2.10) includes only inequality constraints, and we remark that equality constraints can be incorporated with a straightforward modification since they are always included in the optimal active set.

Let X be a closed polytopic set of parameters, defined by $X = \{x \in \mathbb{R}^n \,|Ax \le b\}$. In multi-parametric programming, it is of interest to characterize the solution or solutions of the mp-NLP problem (2.9)-(2.10)for the set X. The solution of an mp-NLP problem is a triple ($V^*(x)$,$Z^*(x)$,X_f),where the set of feasible parameters X_f is the set of all $x \in X$ for which the problem (2.9)-(2.10) admits a solution, i.e.

$$X_f = \{x \in X \,|g(z,x) \le 0\} \tag{2.11}$$

the optimal value function $V^* : X_f \to \mathbb{R}$ associates with every $x \in X$ the corresponding optimal value of the problem (2.9)-(2.10).the optimal set $Z^*(x)$ associates to each parameter $x \in X$ the corresponding set of optimizers $Z^*(x) = \{z \in \mathbb{R}^s \,|\, f(z,x) = V^*(x)\}$ of problem (2.9)-(2.10). If $Z^*(x)$ is a singleton for all $x \in X$, then $z^*(x) \triangleq Z^*(x)$ is called the optimizer function.

2.5 Multiparametric Mixed Integer Programming

Multiparametric mixed integer linear programming (mp-MILP) problems involving (i) 0-1 integer variables, and, (ii) more than one parameter, bounded between lower and upper bounds, present on the right hand side (RHS) of constraints.The solution is approached by decomposing the mp-MILP into two subproblems and then iterating between them. The first subproblem is obtained by fixing integer variables, resulting in a multiparametric linear programming (mp-LP) problem, whereas the second subproblem is formulated as a mixed integer linear programming (MILP) problem by relaxing the parameters as variables.

A method for solving mpMILP problems is suggested in where the authors develop a branch and bound (B & B) based method to solve the problem. The approach is based upon solving one mpLP at each node of the B & B tree, and as in standard B & B methods, complete enumeration of the integer variables is avoided by maintaining upper bounds on the value function. Another solution strategy was developed, in which a geometric approach is followed to avoid solution of the mpLPs at the nodes of the B & B tree.

Consider an mp-MILP problem of the following form:

$$V^*(x) = \min_z c^T z \tag{2.12}$$

$$s.t. \quad Az \leq b + Sx \tag{2.13}$$

where $z \in \mathbb{R}^n$ is the optimization variable, $x \in \mathbb{R}^n$ is the vector of parameters and $c \in \mathbb{R}^{s \times s}$, $A \in \mathbb{R}^{q \times s}$, $b \in \mathbb{R}^q$, and $S \in \mathbb{R}^{q \times n}$ are matrices. The mpMILP is solved by decomposing the problem into mpLP and an MILP subproblems, and propagating through the parameter space in a geometrical fashion.This geometric approach has the advantage of being relatively simple to implement, and has been successfully applied for other problems than mpMILP. If the cost function (2.12) had been a quadratic function in z and x, the problem would have been a multiparametric mixed integer QP (mpMIQP). As exemplified in this geometric approach can, if used to solve an mpMIQP, lead to non-convex regions, and would require non-convex optimization problems to be solved, which of course is undesirable.

2.6 Notation

The notation of the thesis is consistent with the following exception: The notation in the mpQP problem formulation is different in Chapter 3 and Chapter 5. In Chapter 2 the mpQP is defined as

$$V^*(x) = \min_z \frac{1}{2} z^T H z \tag{2.14}$$

$$s.t \quad Gz \leq W + Sx \tag{2.15}$$

where z is the optimization variable and x is the parameter vector. In Chapter 5 the mpQP is defined as

$$V^*(x) = \min_z \frac{1}{2} z^T H z + x^T F^T z + c^T z \tag{2.16}$$

$$s.t. \quad A_i z = b_i + S_i x, \quad i \in \varepsilon \tag{2.17}$$

$$A_i z \leq b_i + S_i x, \quad i \in \kappa \tag{2.18}$$

where z is the optimization variable and x is the parameter vector. The reason for this change of notation is that the paper which Chapter 3 is based on takes the point of view from MPC, in which z is commonly used as the system state, which is also the parameter vector. Chapter 5 takes a more mathematical point of view, and the notation used is similar to what is common when formulating a mathematical program.

CHAPTER 3

An Algorithm for mp-QP and Explicit MPC solutions

3.1 Introduction

Our motivation for investigating multi-parametric quadratic programming (mp-QP) comes from linear model predictive control (MPC). This generates to a class of control algorithms that compute a manipulated variable trajectory from a linear process model to minimize a quadratic performance index subject to linear constraints on a prediction horizon. The first control input is then applied to the process. At the next sample, measurements are used to update the optimization problem and the optimization is repeated. In this way, this becomes a closed loop approach. There has been some limitation to which processes MPC could be used on due to the computationally expensive on-line optimization which was required. There has recently been derived explicit solutions to the constrained MPC problem, which could increase the area of use for this kind of controllers. Explicit solutions to MPC problems are not mainly intended to replace traditional implicit MPC, but rather to extend its area of use. MPC functionality can with this be applied to applications with sampling rates in the micro-second range, using low cost embedded hardware. Software complexity and reliability is also improved, allowing the approach to be used on safety critical applications.

In this work we present an algorithm for the solution of multi-parametric linear and quadratic programming problems.With linear constraints and linear or convex quadratic objective functions, the optimal solution of these optimization problems is given by a conditional piecewise linear function of the varying parameters. This function results from first-order estimations of the analytical non-linear optimal function [20]. The core idea of the algorithm is to approximate the analytical non-linear function by affine functions, whose validity is confined to regions of feasibility and optimality. Therefore, the space of parameters is systematically characterized into different regions where the optimal solution is an affine function of the parameters. The solution obtained is convex and continuous. Examples are presented to illustrate the algorithm and to enhance its potential in real-life applications [18].

3.2 Model Predictive Control

Model Predictive Control(MPC) is a control algorithm based on solving a finite horizon open-loop optimization problem at each sampling instant. Such controller rely on an internal dynamic model of the process used to predict the behaviour of the system. The system to be controlled is usually described by one or more ordinary differential equations. Because MPC is a discrete algorithm, the ordinary differential equations are usually converted to discrete difference equations. The MPC objective cost function is often on the form

$$V(k) = \sum_{t=1}^{i} Q\left(t\right)\left(\hat{x}\left(k+t\left|k\right.\right) - r\left(k+t\left|k\right.\right)\right)^2 + R\left(t\right)\left(\hat{u}\left(k+t\left|k\right.\right)\right)^2 \tag{3.1}$$

Where $\hat{x}$ is the estimation state. r is the reference trajectory. $\hat{u}$ is the optimal control sequence and i is the predictive horizon length. The first term in $V(k)$ represents that the state x should track the reference r. The various states are weighed with $Q(t)$ to reflect relative tracking importance between states. The second term in the cost function will penalize the use of control input u, with weighing vectors $R(t)$. The main advantage of MPC is its ability to handle constraints. Both input constraints (bounds on u), like the saturation of an actuator and state constraints (bounds on x), like keeping the level of a fluid between bounds, can be handled with ease.

The system model is initialized with the most recent sample of the states and the controller uses the combination of these and the internal model to optimize the objective cost function such that the cost is minimized and all constraints are honoured. The controller will only use the first step of the calculated control sequence as plant input. This optimization based approach is the main difference from conventional control strategies, where a precomputed control law is usually applied for each sample time. The basic of MPC are displayed in figure (3.1).

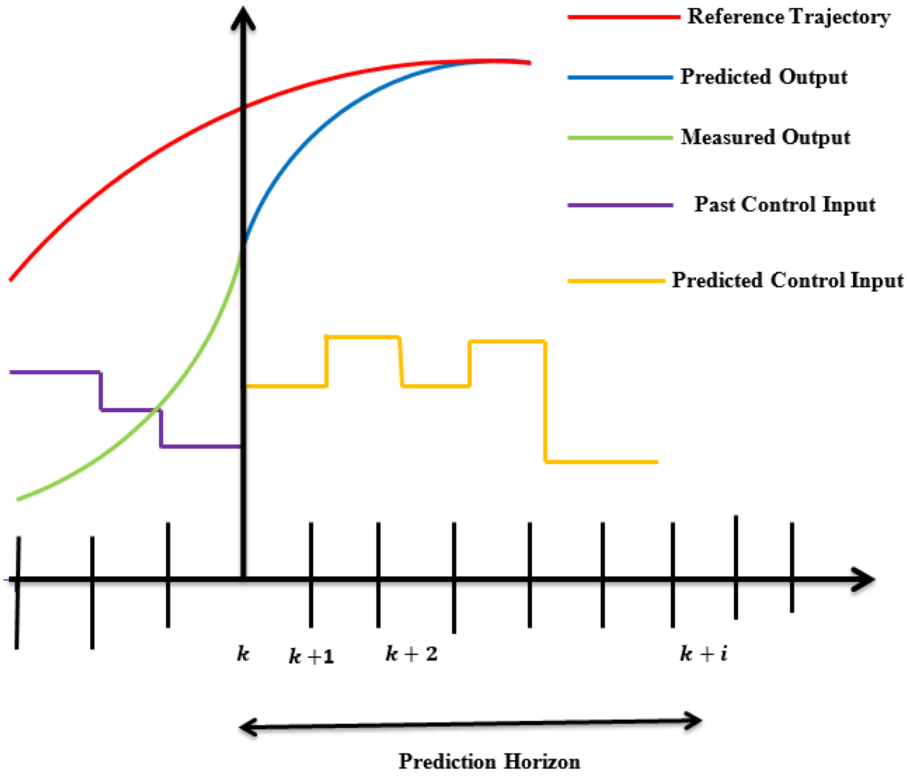

Figure 3.1: A discrete MPC scheme

An explanation of figure (3.1). At time k the current plant state is sampled. The cost function is minimized while honouring constraints, leading to a optimal control strategy for the horizon interval $[k, k+i]$. The predicted optimal output is the blue line which converges towards the red reference, like reflected in the cost function (3.1). The optimal control input is shown in orange.

The control strategy explores state trajectories emanating form the sampled starting point and finds the one minimizing cost. Only the first control step is applied to the plant and the plant state is then sampled again and the same procedure is repeated, giving a new control step and a new predicted state path. Because the horizon keeps beeing pushed forward, MPC is sometimes called receding horizon control (RHC).

The way MPC handles constraints allows for plant operation closer to the optimal working point. It has been widely applied in the chemical and petroleum industries because accounting for constraints is especially important in the these applications. The MPC strategy is also expected to behave well in a control allocation perspective, because of its predictive nature and ability to handle actuator dynamics. Given an estimate of the control allocated craft's future trajectory, it enables the craft to utilize actuators with different time constants to their full extent. This also opens possibilities to restrict the use of costly actuators when not necessary. This cost can be either connected to e.g. a power/fuel consumption or radar cross section concern. For a detailed description of Model Predictive Control. see []

3.3 Using MP-QP method for two predictive state

3.3.1 From Linear MPC to an MpQP Problem

Consider the linear time variant system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{3.2}$$

Where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the input vector. $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{l\times n}$ are system matrix, input and output matrix respectively.For current $x(t)$, the MPC solves the optimization problem

$$\min_U \left\{ J(U, x(t)) = x^T_{t+N_y|t} P x_{t+N_y|t} + \sum_{k=0}^{N_y-1} x^T_{t+k|t} Q x_{t+k|t} + u^T_{t+k} R u_{t+k} \right\} \tag{3.3}$$

$$\begin{aligned} s.t \quad & y_{\min} \le y_{t+k|t} \le y_{\max} \qquad k = 1, \ldots.., N_c \\ & u_{\min} \le u_{t+k} \le u_{\max} \qquad k = 0, \ldots..N_c - 1 \\ & x_{t|t} = x(t) \\ & x_{t+k+1|t} = Ax_{t+k|t} + Bu_{t+k} \qquad k \ge 0 \\ & y_{t+k|t} = Cx_{t+k|t} \qquad k \ge 0 \\ & u_{t+k} = Kx_{t+k|t} \qquad N_c \le k \le N_y \end{aligned}$$

Where $x_{t+k|t}$ refer as the predictive state vector at the $t+k$ and $k = 0, 1$. We assume that $R = R^T > 0$, $Q = Q^T > 0$, $P = P^T > 0$ and $U^* = \{u^*_t, \ldots\ldots u^*_{t+k-1}\}$. N_u, N_y and N_c are the input, output, and constraint horizon respectively, such that $N_y \ge N_u$ and $N_c \le N_y - 1$ and

K is a stabilizing state feedback gain is solved repetitively.

Introducing the following equation, which is derived from (3.2)

$$x_{t+k|t} = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{t+k-1-j} \tag{3.4}$$

And put the equation(3.4) in (3.3) and the results in the following quadratic programming or QP problem

$$\begin{aligned} & V^*(x_t) = \min_U \left\{ \frac{1}{2} U^T H U + x_t^T F U + \frac{1}{2} x_t^T Y x_t \right\} \\ & GU \leq W + S x_t \end{aligned} \tag{3.5}$$

Where $H = H^T \succ 0$ and H,F,Y,G,W and E are obtained from Q, R.

Before we applying multi-parametric quadratic programming method in (3.5), we have to consider the following linear transformation

$$z = U + H^{-1} F^T x_t \tag{3.6}$$

The QP problem (3.5) is then formulated to the following multi-parametric quadratic programming (mp-QP) problem:

$$\begin{aligned} V_z(x_t) &= \min_z \frac{1}{2} z^T H z \\ s.t. \quad Gz &\leq W + S x_t \end{aligned} \tag{3.7}$$

where $z \in \mathbb{R}^s$ is the vector of optimization variable, x_t is the vector of parameters, $S = E + GH^{-1}F^T$ and $V_z(x_t) = V(x_t) - \frac{1}{2} x_t^T (Y - FH^{-1}F^T) x_t$. In the transformed problem, the parameter vector x_t appears only on the rhs of the constraints.

In order to start solving the mp-QP problem, an initial vector x_0 inside the polyhedral set X of parameters is needed, such that the QP problem (3.7) is feasible for $x = x_0$. A good choice for x_0 is the center of the largest ball contained in X for which a feasible z exists. So determined by solving the LP problem:

$$\begin{aligned} & \max_{x,z,\varepsilon} \ (\varepsilon) \\ s.t. \quad & T^i x + \varepsilon \left\| T^i \right\| \leq Z^i \\ & Gz - Sx \leq W \end{aligned} \tag{3.8}$$

where x_0 will be the Chebychev center of X when the QP problem (3.7) is feasible for such an x_0. If $\varepsilon \leq 0$ then the QP problem (3.7) is infeasible for all x in the interior of X. Otherwise, we fix $x = x_0$ and solve the QP problem (3.7), in order to obtain the corresponding optimal solution z_0. That solution is unique, because $H \succ 0$, and therefore uniquely determines a set of active constraints $\tilde{G}z_0 = \tilde{S}x_0 + \tilde{W}$ out of the constraints in QP problem (3.7).

3.3.2 Background on MpQP

Theorem 3.1. *[21] Let $z_0 \in \mathbb{R}^n$ be a vector of parameters and (z_0, λ_0) be a KKT pair for (3.7), where $\lambda_0 = \lambda_0(x_0)$ is a vector of nonnegative Lagrange multipliers, λ, and $z_0 = z(x_0)$ is feasible in (3.7). Also assume that the (i) linear independence constraint satisfaction and (ii) strict complementary slackness conditions hold. Then, there exists in the neighbourhood of x_0 a unique, once continuously differentiable function $[z(x), \lambda(x)]$ where $z(x)$ is a unique isolated minimizer for (3.7) and*

$$\begin{pmatrix} \frac{dz(x)}{dx} \\ \frac{d\lambda(x)}{d\lambda} \end{pmatrix} = -(M_0)^{-1} N_0 \tag{3.9}$$

where

$$M_0 = \begin{pmatrix} H & G_1^T & \cdots & G_q^T \\ -\lambda_1 G_1 & -V_1 & & 0 \\ \vdots & & \ddots & \vdots \\ -\lambda_p G_q & 0 & \cdots & -V_q \end{pmatrix}$$
$$N_0 = \begin{pmatrix} Y & \lambda_1 G_1 & \cdots & \lambda_p G_p \end{pmatrix}^T$$

where G_i denotes the ith row of G, S_i denotes the ith row of S, $V_i = G_i z_0 - W_i - S_i x_0$, W_i denotes the ith row of W, and Y is a null matrix of dimension $(s \times n)$.

The optimization variable $z(x)$ can then be obtained as an affine function of the state x_t by exploiting the first-order KarushKuhn Tucker (KKT) conditions for (3.7).

Theorem 3.2. *[21] Let x be a vector of parameters and assume that assumptions (i) linear independence constraint satisfaction and (ii) strict complementary slackness conditions hold. Then, the optimal z and the associated Lagrange multipliers λ are affine functions of x.*

The first-order KKT conditions for the mp-QP (3.7) are given by

$$Hz + G^T \lambda = 0 \tag{3.10}$$

$$\lambda_i \left(G_i z - W_i - S_i x\right) = 0, \qquad i = 1, \cdots, q \tag{3.11}$$

$$\lambda \geq 0 \tag{3.12}$$

H is invertible (3.10) is written as

$$z = -H^{-1} G^T \lambda \tag{3.13}$$

Let $\breve{\lambda}$ and $\tilde{\lambda}$ denote the Lagrange multipliers corresponding to inactive and active constraints, respectively. For inactive constraints, $\breve{\lambda} = 0$. For active constraints,

$$\tilde{G} z - \tilde{W} - \tilde{S} x = 0 \tag{3.14}$$

where $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ correspond to the set of active constraints. From (3.10)-(3.13),

$$\tilde{\lambda} = -\left(\tilde{G} H^{-1} \tilde{G}^T\right)^{-1} \left(\tilde{W} + \tilde{S} x\right) \tag{3.15}$$

Note that $\left(\tilde{G} H^{-1} \tilde{G}^T\right)^{-1}$ exists because of the linear independence constraint satisfaction assumption. Thus λ is an affine function of x. We can substitute (3.15) into (3.11) to obtain

$$z = H^{-1} \tilde{G}^T \left(\tilde{G} H^{-1} \tilde{G}^T\right)^{-1} \left(\tilde{W} + \tilde{S} x\right) \tag{3.16}$$

and note that z is also an affine function of x.

An interesting observation, resulting from Theorems 1 and 2, is given in the next Theorem.

Theorem 3.3. *[21] Let x_0 be a vector of parameter values and (z_0, λ_0) a KKT pair, where $\lambda_0 = \lambda(x_0)$ is a vector of non-negative Lagrange multipliers, λ, and $z_0 = z(x_0)$ is feasible in (3.7). Also assume that (i) linear independence constraint qualification and (ii) strict complementary slackness conditions hold. Then,*

$$\begin{bmatrix} z(x) \\ \lambda(x) \end{bmatrix} = -(M_0)^{-1} N_0 \left(x - x_0\right) + \begin{bmatrix} z_0 \\ \lambda_o \end{bmatrix} \tag{3.17}$$

where

$$M_0 = \begin{pmatrix} H & G_1^T & \cdots & G_q^T \\ -\lambda_1 G_1 & -V_1 & & 0 \\ \vdots & & \ddots & \vdots \\ -\lambda_p G_q & 0 & \cdots & -V_q \end{pmatrix}$$

$$N_0 = \begin{pmatrix} Y & \lambda_1 G_1 & \cdots & \lambda_p G_p \end{pmatrix}^T$$

where G_i denotes the ith row of G, S_i denotes the ith row of S, $V_i = G_i z_0 - W_i - S_i x_0$, W_i denotes the ith row of W, and Y is a null matrix of dimension $(s \times n)$.

The solution z_0,λ_0 are derived from Theorems 2 and 3 for a specific vector of parameters x_0. We can obtain the solution $z(x)$,$\lambda\,(x)$ for any parameter vector x from (3.17). Therefore the optimization variable z and the control law U are linear, piece-wise affine functions of the state x,$z(x)$ and $U(x)$. In this way the sequence of control law is obtain as an explicit function of the parameter x.

The set of x where solution (3.17) remains optimal is defined as the critical region (CR^0) and can be obtained as follows. Let (CR^R) represent the set of inequalities obtained (i) by substituting $z(x)$ into the inactive constraints in (3.7), and (ii) from the positivity of the Lagrange multipliers corresponding to the active constraints, as follows:

$$CR^R = \left\{ \breve{G} z\,(x) \leq \breve{W} + \breve{S} x(t), \tilde{\lambda}(x) \geq 0 \right\} \tag{3.18}$$

then by removing the redundancy inequalities from (CR^R), we got the (CR^0) as follows:

$$CR^0 = \Delta \left\{ CR^R \right\} \tag{3.19}$$

Where Δ is an operator which removes the redundancy constraints. Then we representation of (CR^0) in the x-space and represents the largest set $x \in X$ such that the combination of the active constraints at the minimizer remains unchanged. Once the critical region (CR^0) has been defined, then the rest of the region $CR^{rest} = X - CR^0$ has to be explored and new critical regions generated. The Theorem 3.4 define the how to explored the rest of the space. Within the closed polyhedral regions CR^0 in X_f the solution $z(x)$ is affine (3.16). The boundary between two regions belongs to both closed regions because the optimum is unique the solution must be continuous across the boundary.

An algorithm for the solution of an mp-QP of the form given in (3.7) to calculate U as an affine function of x and characterize X by a set of polyhedral regions, CRs, is summarized in algorithm. The optimal control sequence $U^*(x)$, once $z(x)$ is obtained by (3.17), is obtained from (3.6).

$$U^*\left(x\right) = \mathrm{z}\left(\mathrm{x}\right) - H^{-1}F^T x \tag{3.20}$$

Finally, the feedback control law

$$\mathrm{u_t} = \left[I \;\; 0 \;\; 0 \,.. \,. \;\; 0\right] \; U^*\left(x_t\right) \tag{3.21}$$

is applied to the process system.

Algorithm 1 (mp-QP solver)

Step 1. For a given space of x solve (3.7) by treating x as a free variable and obtain $[x_0]$.

Step 2. In (3.7) fix $x = x_0$ and solve (3.7) to obtain $[z_0, \lambda_0]$.

Step 3. Obtain $[z(x), \lambda(x)]$ from 3.17.

Step 4. Define CR^R as given in (3.18).

Step 5. From CR^R remove redundant inequalities and define the region of optimality CR^0 as given in (3.19).

Step 6. Define the rest of the region, $CR^{rest} = X - CR^0$.

Step 7. If no more regions to explore, go to the next step, otherwise go to Step 1.

Step 8. Collect all the solutions and unify a convex combination of the regions having the same solution to obtain a compact representation.

The next Theorem define the how to explored the rest of the space.

Theorem 3.4. *Let $X \in \mathbb{R}^n$ be a polyhedron, and $CR^0 = \{x \in X \,|Ax \le b\}$ a polyhedral subset of X, $CR^0 \ne \phi$. Also let*

$$R_i = \left\{ x \in X \left| \begin{array}{c} A^i x > b^i \\ A^j x \le b^j, \forall j < i \end{array} \right. \right\}, i = 1, \cdots, m \tag{3.22}$$

where $m = dim(b)$, and let $CR^{rest} \triangleq \bigcup\limits_{i=1}^{m} R_i$. Then (i)$CR^{rest} \cup CR^0 = X$, (ii)$CR^0 \cap R_i = \phi$,$R_i \cap R_j = \phi, \forall j \ne i$, i.e. $\left\{CR^0, R_1, \cdots, R_m\right\}$ is a partition of X.

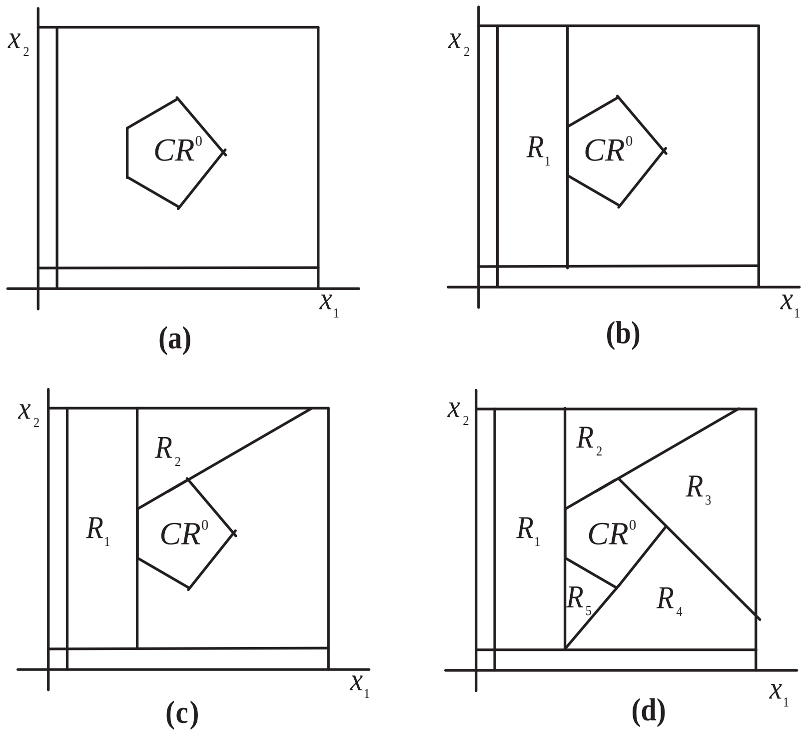

Figure 3.2: (a) Partition of $CR^{rest} \triangleq X \backslash CR^0$; (b) partition of CR^{rest} step 1; (c) partition of CR^{rest} step 2; (d) final partition of CR^{rest}

Theorem 3.5. *For the mp-QP problem (3.7), the set of feasible parameters $X_f \subseteq X$ is convex, the optimal solution, $z(x) : X_f \mapsto \mathbb{R}^s$ is continuous and piecewise affine, and the optimal objective function $V_z(x) : X_f \mapsto \mathbb{R}$ is continuous, convex, and piecewise quadratic.*

Proof: Consider the parameter $x_1, x_2 \in X_f$ and $V_z(x_1)$, $V_z(x_2)$ are the optimal value. Let $z_1,\ z_2$ be the minimizers parameter. Here we have to proof convexity of X_f and $V_z(x)$. Define the equation $z_\alpha \triangleq \alpha z_1 + (1-\alpha) z_2$, $x_\alpha \triangleq \alpha x_1 + (1-\alpha) x_2$. By feasibility, the constraints are $Gz_1 \leq W + Sx_1,\ Gz_2 \leq W + Sx_2$ satisfy the minimizer parameter z_1,z_2. These inequalities can be linearly combined to obtain $Gz_\alpha \leq W + Sx_\alpha$ and therefore z_α is feasible for the optimization problem (3.7) where $x_t = x_\alpha$. Since a feasible solution $z(x_\alpha)$ exists at x_α, an optimal solution exists at x_α and hence X_f is convex. The optimal solution at x_α will be less than or equal to the feasible solution, i.e

$$V_z(x_\alpha) \leq \frac{1}{2} z_\alpha^T H z_\alpha$$

and hence

$$V_z(x_\alpha) - \frac{1}{2}\left[\alpha z_1^T H z_1 + (1-\alpha) z_2^T H z_2\right]$$
$$\leq \frac{1}{2} z_\alpha^T H z_\alpha - \frac{1}{2}\left[\alpha z_1^T H z_1 + (1-\alpha) z_2^T H z_2\right]$$

$$= \frac{1}{2}\left[\alpha^2 z_1^T H z_1 + (1-\alpha)^2 z_2^T H z_2 + 2\alpha(1-\alpha) z_2^T H z_1 - \alpha z_1^T H z_1 - (1-\alpha) z_2^T H z_2\right]$$
$$= -\frac{1}{2}\alpha(1-\alpha)(z_1 - z_2)^T H (z_1 - z_2) \leq 0$$

i.e.

$$V_z\left(\alpha x_1 + (1-\alpha) x_2\right) \leq \alpha V_z\left(x_1\right) + (1-\alpha) V_z\left(x_2\right)$$

for all $x_1, x_2 \in X$. Where $\alpha \in [0,1]$, which proves the convexity of $V_z(x)$ on X_f.

3.4 Numerical Example for two state predictive

Consider the state space representation

$$\mathrm{x}_{t+1} = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} \mathrm{x}_t + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} \mathrm{u}_t$$
$$\mathrm{y}_t = \begin{bmatrix} 0 & 1.4142 \end{bmatrix} \mathrm{x}_t$$

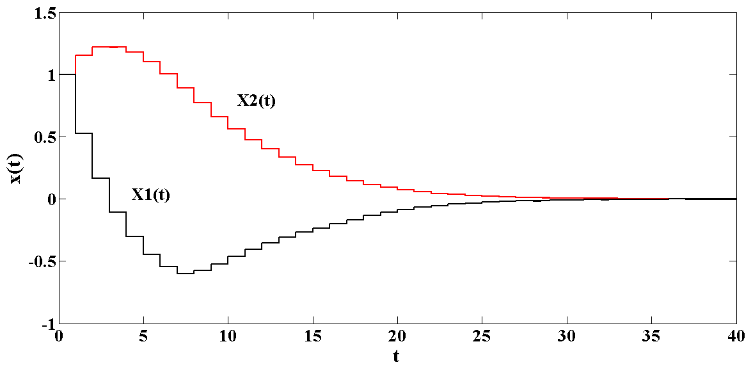

Figure 3.3: State diagram of closed-loop MPC

The constrains on input are $-2 \leq \mathrm{u}_t \leq 2$. The corresponding optimization problem for

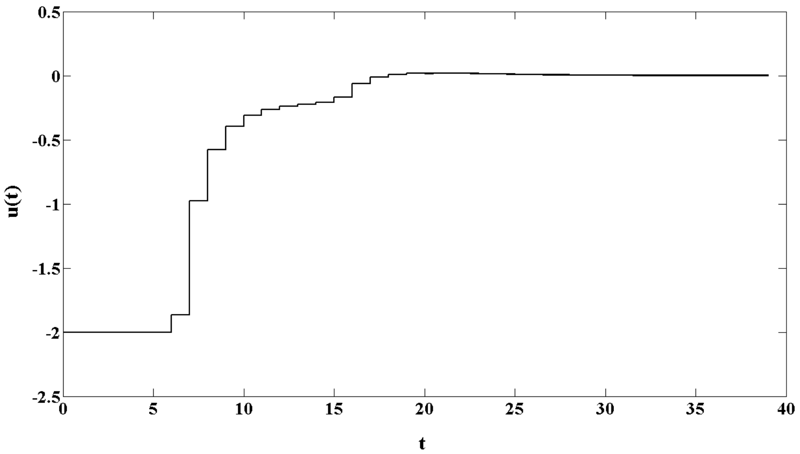

Figure 3.4: optimal control(u) diagram of closed-loop MPC

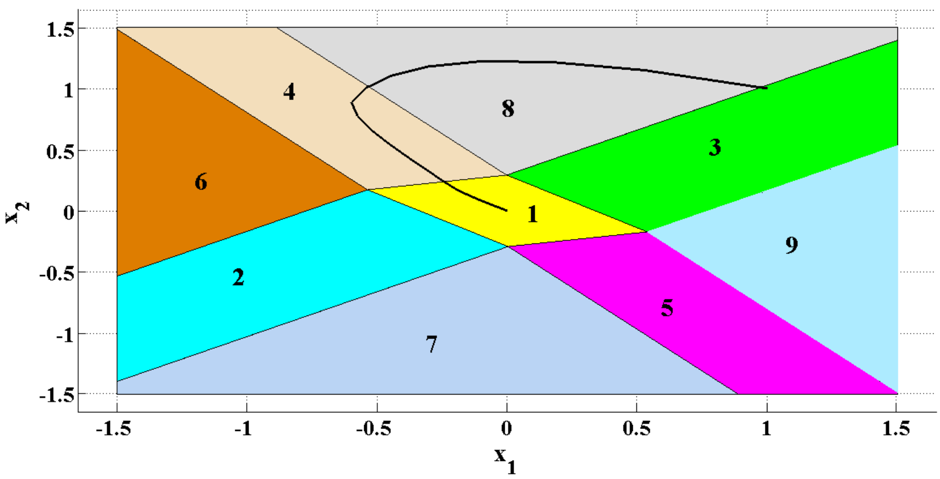

Figure 3.5: State space partition and closed-loop MPC trajectories diagram

regulating to the origin is given

$$\min_{\mathrm{u_t, u_{t+1}}} \mathrm{x'_{t+2|t} x_{t+2|t}} + \sum_{\mathrm{k=0}}^{1} \mathrm{x'_{t+k|t} x_{t+k|t}} + 0.01\mathrm{u^2_{t+k}}$$

$$s.t. \quad -2 \leq \mathrm{u_{t+k}} \leq 2, \qquad k = 0, 1$$

Where P solves the Lyapunov equation $P = A^t P A + Q$

$$P = \begin{bmatrix} 3.0485 & -2.5055 \\ -2.5055 & 12.9916 \end{bmatrix} \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad R = 0.01$$

$$N_u = N_y = N_c = 2$$

Table 3.1: Parametric solution of the numerical example for two state predictive

Region No	Region	control law
1	$\begin{bmatrix} -5.9302 & -6.8985 \\ 5.9302 & 6.8985 \\ -1.5347 & 6.8272 \\ 1.5347 & -6.8272 \end{bmatrix} x \leq \begin{bmatrix} 2.000 \\ 2.000 \\ 2.000 \\ 2.000 \end{bmatrix}$	$\begin{bmatrix} -5.9302 & -6.8985 \end{bmatrix} x$
2	$\begin{bmatrix} -3.4121 & 4.6433 \\ 3.4121 & -4.6433 \\ 0.1044 & 0.1215 \end{bmatrix} x \leq \begin{bmatrix} 2.6331 \\ 1.3669 \\ -0.0352 \end{bmatrix}$	2.000
3	$\begin{bmatrix} -3.4121 & 4.6433 \\ 3.4121 & -4.6433 \\ -0.1044 & -0.1215 \end{bmatrix} x \leq \begin{bmatrix} 1.3669 \\ 2.6331 \\ -0.0352 \end{bmatrix}$	-2.000
4	$\begin{bmatrix} -6.4235 & -4.7040 \\ 6.4235 & 4.7040 \\ 0.0274 & -0.1220 \end{bmatrix} x \leq \begin{bmatrix} 2.6429 \\ 1.3571 \\ -0.0357 \end{bmatrix}$	$\begin{bmatrix} -6.4159 & -4.6953 \end{bmatrix} x - 0.6423$
5	$\begin{bmatrix} -6.4235 & -4.7040 \\ 6.4235 & 4.7040 \\ -0.0274 & 0.1220 \end{bmatrix} x \leq \begin{bmatrix} 1.3571 \\ 2.6429 \\ -0.0357 \end{bmatrix}$	$\begin{bmatrix} -6.4159 & -4.6953 \end{bmatrix} x + 0.6423$
6	$\begin{bmatrix} 0.1259 & 0.0922 \\ 0.0679 & -0.0924 \end{bmatrix} x \leq \begin{bmatrix} -0.0518 \\ -0.0524 \end{bmatrix}$	2.000
7	$\begin{bmatrix} 0.1259 & 0.0922 \\ -0.0679 & 0.0924 \end{bmatrix} x \leq \begin{bmatrix} -0.0266 \\ -0.0272 \end{bmatrix}$	2.000
8	$\begin{bmatrix} -0.1259 & -0.0922 \\ 0.0679 & -0.0924 \end{bmatrix} x \leq \begin{bmatrix} -0.0266 \\ -0.0272 \end{bmatrix}$	-2.000
9	$\begin{bmatrix} -0.1259 & -0.0922 \\ -0.0679 & 0.0924 \end{bmatrix} x \leq \begin{bmatrix} -0.0518 \\ -0.0524 \end{bmatrix}$	-2.000

The MPC problem convert to mp-QP form

$$H = \begin{bmatrix} 0.0196 & 0.0063 \\ 0.0063 & 0.0199 \end{bmatrix} \quad F = \begin{bmatrix} 0.1259 & 0.0679 \\ 0.0922 & -0.0924 \end{bmatrix} \quad G = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}$$

$$W = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \quad E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad S = \begin{bmatrix} 5.9302 & 6.8985 \\ -5.9302 & -6.8985 \\ 1.5347 & -6.8272 \\ -1.5347 & 6.8272 \end{bmatrix}$$

The solution of the mp-QP problem, as computed by using the algorithm and is depicted in figure. To illustrate how on-line optimization reduces to a function evaluation.The solution of the linear MPC optimization problem, with a quadratic objective and linear output and input constraints, by using multi-parametric programming techniques and specifically multi-parametric quadratic programming, provides a complete map of the optimal control as a function of the states and the characteristic partitions of the state space where this solution is feasible.

In that way the solution of the MPC problem is obtained as piecewise affine feedback control law. The on-line computational effort is small since the on-line optimization problem is solved off-line and no optimizer is ever called on-line. In contrast, the on-line optimization problem is reduced to a mere function evaluation problem; when the measurements of the state are obtained and the corresponding region and control action are obtained by evaluation of a number of linear inequalities and a linear affine function, respectively. This is known as the on-line optimization via off-line parametric optimization concept.

3.5 Numerical Example for three state predictive

Consider the state space representation

$$\mathrm{x}_{t+1} = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} \mathrm{x}_t + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} \mathrm{u}_t$$

$$\mathrm{y}_t = \begin{bmatrix} 0 & 1.4142 \end{bmatrix} \mathrm{x}_t$$

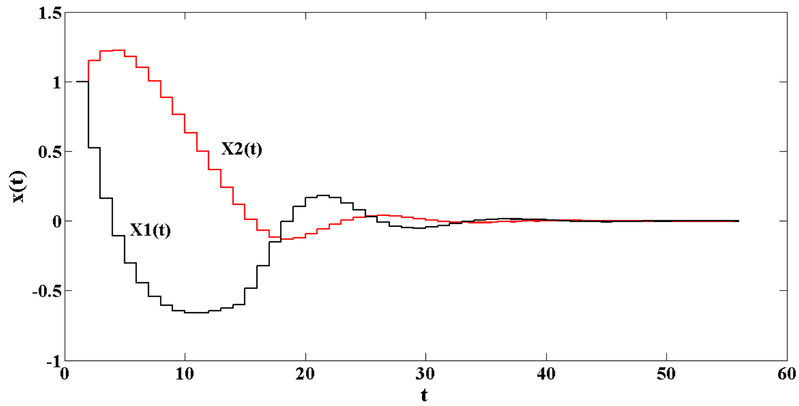

Figure 3.6: State diagram of closed-loop MPC

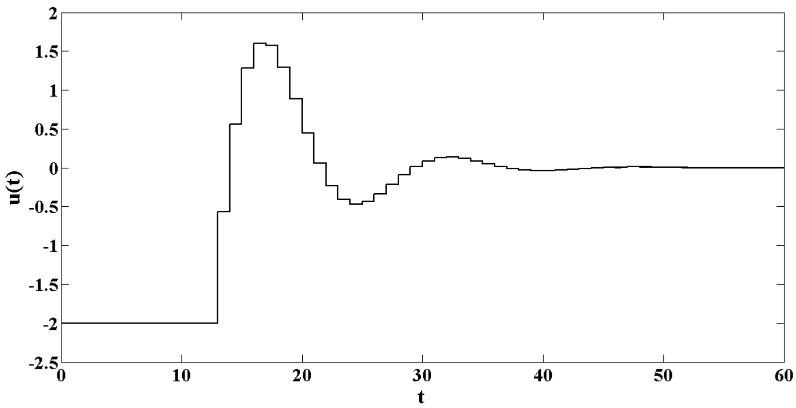

Figure 3.7: optimal control (u) diagram of closed-loop MPC

The constrains on input are $-2 \leq \mathrm{u_t} \leq 2$. The corresponding optimization problem for regulating to the origin is given

$$\min_{\mathrm{u_t, u_{t+1}}} \mathrm{x'_{t+2|t} x_{t+2|t}} + \sum_{\mathrm{k=0}}^{1} \mathrm{x'_{t+k|t} x_{t+k|t}} + 0.01\mathrm{u^2_{t+k}}$$

$$s.t. \quad -2 \leq \mathrm{u_{t+k}} \leq 2, \qquad k = 0, 1, 2$$

Where P solves the Lyapunov equation $P = A^t P A + Q$

$$P = \begin{bmatrix} 3.0485 & -2.5055 \\ -2.5055 & 12.9916 \end{bmatrix} \qquad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad R = 0.01$$

$$N_u = N_y = N_c = 3$$

Table 3.2: Parametric solution of the numerical example for three state predictive

Region No	Region	control law
1	$\begin{bmatrix} -1.3901 & -11.8477 \\ -1.3165 & 7.0156 \\ 1.3901 & 11.8477 \\ 1.3165 & -7.0156 \end{bmatrix} x \leq \begin{bmatrix} 2.000 \\ 2.000 \\ 2.000 \\ 2.000 \end{bmatrix}$	$\begin{bmatrix} -1.3901 & -11.8477 \end{bmatrix} x$
2	$\begin{bmatrix} 0.0353 & 0.2421 \\ -0.0304 & 0.0999 \end{bmatrix} x \leq \begin{bmatrix} -0.0382 \\ -0.0329 \end{bmatrix}$	2.000
3	$\begin{bmatrix} 0.0304 & -0.0999 \\ -0.0353 & -0.2421 \end{bmatrix} x \leq \begin{bmatrix} -0.0329 \\ -0.0382 \end{bmatrix}$	-2.000
4	$\begin{bmatrix} -1.5447 & 5.0709 \\ 1.5447 & -5.0709 \\ -0.0253 & -0.2585 \end{bmatrix} x \leq \begin{bmatrix} 1.6717 \\ 2.3283 \\ -0.0436 \end{bmatrix}$	-2.000
5	$\begin{bmatrix} -1.5806 & -10.8327 \\ 1.5806 & 10.8327 \\ -0.0253 & 0.1349 \end{bmatrix} x \leq \begin{bmatrix} 1.7106 \\ 2.2894 \\ -0.0385 \end{bmatrix}$	$\begin{bmatrix} -1.1996 & -12.8627 \end{bmatrix} x - 0.2893$
6	$\begin{bmatrix} -1.5806 & -10.8327 \\ 1.5806 & 10.8327 \\ 0.0253 & -0.1349 \end{bmatrix} x \leq \begin{bmatrix} -2.2893 \\ -1.7107 \\ -0.0385 \end{bmatrix}$	$\begin{bmatrix} -1.1996 & -12.8627 \end{bmatrix} x + 0.2893$
7	$\begin{bmatrix} -1.5447 & 5.0709 \\ 1.5447 & -5.0709 \\ 0.0303 & 0.2585 \end{bmatrix} x \leq \begin{bmatrix} -2.3283 \\ -1.6717 \\ -0.0436 \end{bmatrix}$	2.000

The MPC problem convert to mp-QP form

$$H = \begin{bmatrix} 0.0227 & 0.0083 & 0.0035 \\ 0.0083 & 0.0196 & 0.0063 \\ 0.0035 & 0.0063 & 0.0199 \end{bmatrix} \qquad F = \begin{bmatrix} 0.0399 & 0.1081 & 0.0339 \\ 0.2455 & 0.0804 & -0.0973 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad W = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \qquad E = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad S = \begin{bmatrix} 1.3901 & 11.8477 \\ 0.4504 & 0.1340 \\ 1.3165 & -7.0156 \\ -1.3901 & -11.8477 \\ -0.4504 & -0.1340 \\ -1.3165 & 7.0156 \end{bmatrix}$$

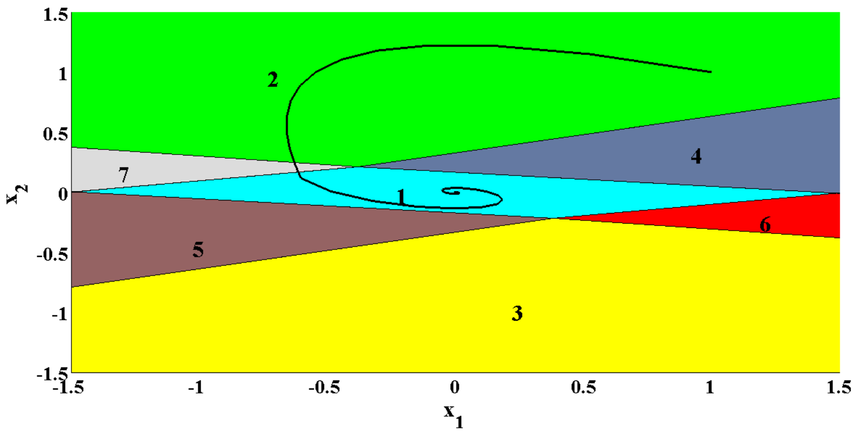

Figure 3.8: State space partition and closed-loop MPC trajectories diagram

The solution of the mp-QP problem, as computed by using the algorithm and despicted in fig.3.8.In this case the control law for each region are different. There has no common control law any two regions and each regions are convex set.

3.6 Conclusion

We have proposed a new approach for solving mp-QP problems giving off-line piecewise affine explicit solutions to MPC control problems. The method is based on the exploitation of direct relations between neighbouring polyhedral regions and combinations of active constraints, and we believe that our contribution significantly advances the field of explicit MPC control, both theoretically and practically, as examples have indicated large improvements of computational efficiency over existing mp-QP algorithms.

CHAPTER 4

Control allocation via mpQP method

4.1 Introduction

Control allocation design for system with effector redundancy is challenging since multiple combinations of the available control effectors can generate the same desired control. In addition to this, actuator constraints should be consider in account. Adding a control allocation module essentially splits the control design into two separate parts : a control law for generating the desired control variables and the control allocation part for the distribution of control power. This has many benefits, some listed in [22] include easy reconfiguration in case of actuator change, separated regulation tuning, and lastly that the control allocation method can be arbitrary. Because of this last fact there exists a lot of different control allocation methods, ranging from simple to complex.

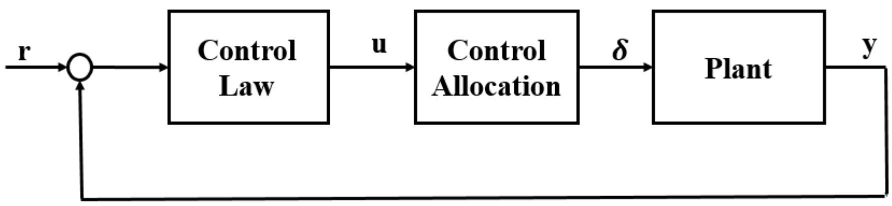

Figure 4.1: Split control configuration

In the classic formulations of the constrained control allocation problem, the actuator dynamics are neglected [16]. This is done under the assumption that the actuator dynamics are orders of magnitude faster than the aircraft dynamics, and can be ignored, or that all dynamic phenomena are accounted for by the controller that commands the virtual control to the control allocation module. In some cases this may be an unrealistic and inconvenient assumption, i.e. when the actuator dynamics are limiting the control performance because response times and different dynamic authorities of the actuators are not taken into account.

Control allocation plays a vital role in ship control systems [23], flight control systems [17] and other over-actuated mechanical control applications [24]. The control allocation module will send control signals to the individual actuators in order to produce the required forces and moments commanded from a higher level control system or pilot during manual operation.

Such over-actuated control allocation problems are naturally formulated as optimization problems as one usually wants to take advantage of all available degrees of freedom in order to minimize power consumption, drag, tear/wear and other costs related to the use of control, subject to constraints such as actuator position limitations [14] [15]. Generally the constrained optimization problems are hard to solve using state-of-theart iterative numerical optimization software at a high sampling rate in a safety-critical real-time system with limiting processing capacity and high demands for software reliability. The main disadvantage are worst case computational complexity and software verification is a complicated issue.

4.2 Basic over view of control allocation

To introduce the ideas behind control allocation, consider the system

$$\dot{x} = u_1 + u_2 \tag{4.1}$$

Where x is a scalar state variable, and u_1 and u_2 are control input, x can be affected by two actuators. Assume that to accelerate the object, the net force $v = 1$ is to be produced. There are several ways to achieve this. We can choose to utilize only the first actuator and select $u_1 = 1$, $u_2 = 0$, or to gang the actuators and use $u_1 = u_2 = 0.5$.

In linear control theory, there is a wide range of control design methods, like LQ design, which perform control allocation and regulation in one step. Thus, the usefulness of control allocation for linear systems is not so obvious. There are however other, more practical reason to use a separate control allocation module, even for linear system. One benefit is that actuator constraints can be taken into account. If one or more actuator saturates, and fail to produce its nominal control effect, another actuator may be used to make up the difference.

Linear equation:

Consider first a linear dynamic system on state space form.

$$\dot{x} = Ax + B_u u \tag{4.2}$$

Where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input, $A \in \mathbb{R}^{n \times n}$ and $B_u \in \mathbb{R}^{n \times n}$. Assume that B_u has rank $k < m$. Then B_u has a null-space of dimension $m - k$ in which we can perturb the control input without affecting x. Thus, there are several choices of control input that gives the same system dynamics. This is the type of redundancy that can be resolved using control .

Since B_u is rank deficient it can be factorized as

$$B_u = B_v B \tag{4.3}$$

Where$B_v \in \mathbb{R}^{n \times k}$ and$B \in \mathbb{R}^{k \times m}$ both have rank k. Introducing the virtual control input

$$v = Bu$$

Where $v \in \mathbb{R}^k$, we can rewrite the systems dynamics (4.2) as

$$\dot{x} = Ax + B_v v \tag{4.4}$$

Now, control design can be performed in two steps, as outlined in the introduction.

Non-linear systems:

The same ideas can be used to deal with non-linear systems of the form

$$\dot{x} = f\left(x, g\left(x, u\right)\right) \tag{4.5}$$

Where $f : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ where $k < m$. Introducing the virtual control input

$$v = g(x, u) \tag{4.6}$$

where $v \in \mathbb{R}^k$, we can rewrite (4.5) as

$$\dot{x} = f(x, u)$$

and again use a two-step control design. A special class of non-linear systems is systems of the form

$$\dot{x} = f(x) + g_u(x, u)$$

$$g_u(x, u) = B_v g(x, u)$$

where $B_v \in \mathbb{R}^{n \times k}$ and f and g are non-linear mappings as above. Again introducing $v = g(x, u)$ yields

$$\dot{x} = f(x) + B_v v$$

Note that these resulting dynamics are affine in v, which simplifies many non-linear design methods like, for example, back-stepping.

Solving (4.6) for u, while considering the actuator constraints $u_{\min} \leq u \leq u_{\max}$, amounts to performing constrained non-linear programming. Since control allocation is to be performed in real time, this may not be computationally feasible. One way to resolve this problem is to approximate (4.6) locally with an affine mapping. Linearising g around u_0 yields

$$g(x, u) \approx g(x, u_0) + \frac{\partial g}{\partial u}(x, u_0) \,.\, (u - u_0) \tag{4.7}$$

where $(x) = \frac{\partial g}{\partial u}(x, u_0)$ Which leads to the linear control allocation problem

$$\bar{v} = B(x)\, u$$

where

$$\bar{v} = v - g(x, u_0) + B(x)\, u_0 \tag{4.8}$$

and methods for linear control allocation can be used.

Direct allocation problem:

Given a matrix B, find a real number a and a vector u_1 such that:

$$\begin{aligned} & J = \max_a a \\ s.t. \quad & (B)\, u_1 = av \\ & u_{min} \leq u \leq u_{max} \end{aligned} \tag{4.9}$$

If $a > 1$, let $u = \frac{u_1}{a}$. Otherwise let $u = u_1$

An advantage of direct allocation includes the straight forwardness of the allocation problem. No design variables must be selected, since the solution to the problem is determined by the control effectiveness matrix (B) and the constraints. When $a > 1$ no element in u will be saturated. A method of implementing direct allocation is by using linear programming.

The objective of direct control allocation is to find a control vector u which gives the best approximation of v in the given direction. Thus direct control allocation weighs directionality over moment generation, which is an important characteristic especially for applications such as flight control. In a special case of the matrix B direct allocation provides a unique solution to the problem. The condition for this property is that any q rows of B must be linearly independent, where q is the number of rows in B. In flight control the case is most often that the rows in B are three. In this case the three components of v in the model reference control law is the accelerations in p, q and r as outputs are three rotational accelerations. The columns of B represent the contributions of the various control surfaces to each of the three rotational accelerations.

4.3 The control allocation problem

Let the consider commanded forces in (x, y, z) be denoted (τ_x, τ_y, τ_z) and the commanded moments in roll, pitch and yaw be denoted $(\tau_\varnothing, \tau_\theta, \tau_\varphi)$. These are stacked in a vector of commanded generalized forces $\tau = (\tau_x, \tau_y, \tau_z, \tau_\varnothing, \tau_\theta, \tau_\varphi)^T$. Assume the system is equipped with N linear actuators with control inputs u_i. If each actuator is characterized by a monotonous non-linearity, it is implicitly assumed that this non-linearity is inverted. The kinematics then leads to a relationship between the controls $u = (u_1, u_2, \ldots\ldots u_N)^T$ and the generalized forces $\tau \in \mathbb{R}^m$ of the following form

$$Bu = \tau \tag{4.10}$$

where $B \in \mathbb{R}^{n \times m}$. In many control allocation applications not all six components of τ are specified. For example, in aerospace applications one is often only concerned with the three body-axis moments, where as in dynamic position applications involving marine surface vessels one is usually concerned only with the three horizontal plane components $\tau = (\tau_x, \tau_y, \tau_\varphi)$.

When constraints are neglected, the common solution to the problem is the generalized inverse, defined as $B^{+} = R^{-1}B^{T}\left(BR^{-1}B^{T}\right)^{-1}$ assuming the configuration is non-singular such that B has full rank.

$$u = B^{*}\tau \tag{4.11}$$

Which solves the least-squares problem

$$\begin{aligned} &\min_{u} u^{T}Ru \\ s.t. \quad &Bu = \tau \end{aligned} \tag{4.12}$$

And $R \in \mathbb{R}^{m \times n}$, where $R > 0$ is a weighting matrix. The most important feature of this approach is that it admits an explicit solution that is computationally efficient and easily implemented. In order to improve robustness near singular configurations, some modifications are suggested in. It is, however, of interest to consider more advanced optimization formulations that allows more general cost indices and in particular considers the presence of constraints on u, as this will in general improve the performance.

$$u_{min} \leq u \leq u_{max} \tag{4.13}$$

where $u_{min}, u_{max} \in \mathbb{R}^{n}$. where the inequalities are to be considered element wise. When taking constraints on u into consideration, one can in general identity two different objectives for the control allocation.

First is control sufficiency. This means that there exists a feasible solution attains the desired generalized force τ. In this case , to minimize some norm of u, to minimize the cost of control

$$\begin{aligned} &\min_{u} \; \|u\| \\ s.t. \quad &Bu = \tau \\ &u_{min} \leq u \leq u_{max} \end{aligned} \tag{4.14}$$

The second is control deficiency. When a feasible u that solves (4.10) does not exist, the difference $Bu - \tau$ should be minimized. The direct control allocation does this by finding a addition that preserves the direction of the generalized force vector, alternatively a norm of this difference is minimized

$$\begin{aligned} &\min_{u} \; \|Bu - \tau\| \\ s.t. \quad &u_{min} \leq u \leq u_{max} \end{aligned} \tag{4.15}$$

The two objectives (4.14) and (4.15) will be combined into a single optimization problem similar to the mixed optimization problem formulated in

$$\min_{u} \frac{1}{2}\left(s^T Q s + u^T R u\right) \tag{4.16}$$

$$s.t. \quad Bu = \tau + s \tag{4.17}$$

$$u_{min} \leq u \leq u_{max} \tag{4.18}$$

Where s is a vector of slack variables used to penalize $Bu - \tau$. Note that by combining the two objectives in this fashion the solution can have a nonzero and even when $\tau\epsilon\mathcal{L}$. The weighting matrix Q should be chosen much larger than R, to prioritize objective 2 to objective 1. Thus $s = 0$ whenever $\tau\epsilon\mathcal{L}$. Note that one can off-line compute the largest value of s for $\tau\epsilon\mathcal{L}$ from the explicit solution by solving a linear program (LP) for each polyhedral region in the explicit solution. If this s is unacceptably large, one should increase the weighting matrix Q.

4.4 Control allocation problem using MPQP

The optimization problem (4.16)-(4.18) can for a given τ be considered as a QP. One could, therefore consider solving this QP for every sample to obtain the optimal solution to problem (4.16)-(4.18). when $z = \left(u^T, s^T\right)$ and $x = \tau$, it is straightforward to see that the above optimization problem can be reformulated as follows:

$$\min_{z} \frac{1}{2} z^T H z \tag{4.19}$$

$$s.t. \quad G_1 z = W_1 + S_1 x \tag{4.20}$$

$$G_2 z = W_2 + S_2 x \tag{4.21}$$

when $H = diag\,(R, Q)$, $G_1 = (B| - I_{n\times m})$, $G_2 = (I_{n\times m}|\, 0_{n\times m}, -I_{m\times m}\; 0_{m\times n})$, $W_1 = 0_{n\times 1}$, $W_2 = \left(u_{max}^T, u_{min}^T\right)^T$, $S_1 = I_{n\times n}$ and $S_2 = 0_{2m}$. Where diag denotes a block-diagonal matrix because $Q > 0$ and $R > 0$ implies $H > 0$, thus this defines a convex quadratic problem in z parametrized by x. It has recently been found that the solution to such problems is a continuous piece-wise linear function $z^*(x)$ and it defined on an polyhedral partition of any polyhedral domain in the parameter space.

4.4.1 Multi-parametric Quadratic Programming

From [1], the mp-QP problem

$$\begin{aligned} V_z(x_t) &= \min_z \frac{1}{2} z^T H z \\ s.t. \quad & Gz \le W + S x_t \end{aligned} \tag{4.22}$$

This can be solved by applying the Karush-Kuhn-Tuker (KKT) conditions

$$Hz + G^T \lambda = 0 \tag{4.23}$$

$$\lambda_i \left(G_i z - W_i - S_i x \right) = 0, \qquad i = 1, \cdots, q \tag{4.24}$$

$$\lambda \ge 0 \tag{4.25}$$

Superscript i on some matrix denotes the i^{th} row. Considering H has full rank, (4.23) gives

$$z = -H^{-1} G^T \lambda \tag{4.26}$$

Assume for the moment that we know which constraints are active at the optimum for a given x, and let $\tilde{\lambda}$ be the Lagrange multipliers of the active constraints, $\tilde{\lambda} \ge 0$. We can now form matrices $\tilde{G}$,$\tilde{W}$ and $\tilde{S}$ which contains the row G^i,W^i and S^i corresponding to the active constraints. Consider that $\tilde{G}$ has full row rank, such that the rows of $\tilde{G}$ are linearly independent. For the active constraints, (4.24) and (4.26) gives $\tilde{G}z - \tilde{W} - \tilde{S}x = 0$, which leads to

$$\tilde{\lambda} = -\left(\tilde{G} H^{-1} \tilde{G}^T \right)^{-1} \left(\tilde{W} + \tilde{S} x \right) \tag{4.27}$$

Equation (4.27) can now be substituted into (4.26) to obtain

$$z = H^{-1} \tilde{G}^T \left(\tilde{G} H^{-1} \tilde{G}^T \right)^{-1} \left(\tilde{W} + \tilde{S} x \right) \tag{4.28}$$

We have now characterized the solution to (4.22) for a given optimal active set, and a fixed x. However, as long as the active set remains optimal in a neighbourhood of x, the solution (4.28) remains optimal, when z is viewed as a function of x. Next, we characterize the region where this active set remains optimal. First, z must remain feasible

$$G H^{-1} \tilde{G}^T \left(\tilde{G} H^{-1} \tilde{G}^T \right)^{-1} \left(\tilde{W} + \tilde{S} x \right) \le W + Sx \tag{4.29}$$

and also the Lagrange multipliers λ must remain non-negative

$$-\left(\tilde{G}H^{-1}\tilde{G}^T\right)^{-1}\left(\tilde{W}+\tilde{S}x\right) \geq 0 \tag{4.30}$$

The inequalities (4.29) and (4.30) describe a polyhedron in the parameter space. This region is describe a polyhedron in the parameter space. This region is denoted as the critical region CR^0 corresponding to the given set of active constraints. This region is a convex polyhedral set and represent set of parameters x such that the combination of active constraints at the minimizer remains optimal.

An algorithm has been developed in [1] for constructing polyhedral partitions of the parameter space that explicitly defines the PWL function $z^*(x)$. Below, we give a simplified description of the algorithm, while a more comprehensive description and analysis that also covers degeneracy and infeasibility is found.

Algorithm 2 (off-line mp-QP solver)

Step 1. Initialize the list of unexplored active sets u with an arbitrary (but feasible) active set. Initialize the first of explored active sets ε to be empty.

Step 2. Choose an arbitrary active set in u, compute the associated linear state feedback (4.28), Lagrange multiplier (4.27) and polyhedral region CR^0 defined by (4.29) and (4.30). Remove the active set under consideration from u and add it to ε.

Step 3. If $CR^0 = \phi$, go to step 2, otherwise go to step 4.

Step 4. For each facet of the corresponding polyhedral representation determine the active set in the neighbouring region as described in detail in (4.22). For each new active set (i.e. not already in $\varepsilon \cup u$), add it to u.

Step 5. If u is non-empty, go to step 2, otherwise terminate.

CHAPTER 5

Conclusion and Future Scope

5.1 Discussion and Conclusion

This thesis has treated theoretical and practical issues in the intersection between multiparametric programming and constrained optimal control. The purpose of this chapter is to give a summary of the main conclusions that can be drawn from the work.

Using explicit solutions to RHC problems by multiparametric programming has clear advantages but also a few drawbacks/limitations. Among the most important advantages are

- The simple structure of the solution (PWL) leads to a real-time implementation which can be made with a few lines in software. This is important in safety-critical applications, as the implementation can be easily verified.

- The implementation can be made on inexpensive hardware, as fixed point arithmetic can be used. This is an important feature in mass-produced equipment, e.g. in the automotive industry.

- The attainable sampling rates are high. This allows RHC functionality for fast (e.g. mechanical) systems with constraints.

But there are also some disadvantages which limit the use of these methods:

- The memory requirements are generally higher for explicit solutions than the case is when using on-line optimization software. Even if some work has been made on simplification/approximation of the explicit solutions, this remains the main limitation for these methods.

- One advantage of the traditional way of using RHC, is that the controller may be easily modified to handle configuration changes, fault conditions etc. This advantage is to some extent lost when using explicit solutions, as the off-line time to construct a new controller may be large. To some extent such situations can be handled by introducing extra parameters into the multi-parametric program, or by a priori generating several controllers for different modes of operation.

- The method is limited to fairly small problems, because memory requirements and off-line computation times seems to increase more or less exponentially with problem dimension.

The PWL control laws obtained from explicit RHC solutions increase rapidly in complexity when the problem size grows. A natural question raised is How can a complex PWL function be represented for efficient and reliable real-time implementation?. One possible answer to this is the binary search tree structure suggested in Chapter 4. When creating such a binary tree, the goal is a tree with low worst case evaluation time, and low memory requirements. An off-line algorithm is proposed, giving a tree with an evaluation time which is logarithmic in the number of regions representing the PWL function. The method is expected to increase the sampling rates to which complex PWL control can be applied.

The second application area treated in this thesis, is constrained control allocation in over-actuated mechanical systems. This is an area particularly well suited for this kind of solutions, as the problem sizes are relatively small, and real-time optimization is often ruled out due to safety reasons. The method is compared to methods in the literature, showing good results, both in terms of optimality of the solution and real-time computational requirements.

5.2 Future Scope

Even if the last few years have given much development within the field of multi-parametric programming within constrained optimal control, improvements can still be made. Among the subjects touched in this thesis, one may consider the following:

- Some work on approximate solutions to mpQPs has been reported in the literature. An inherent feature of parametric program solutions seems, however, to be that the exact solution is easier to characterize than an approximation. Future contributions in this area would be important, as the main limitation of parametric program solutions is the rapid growth of solution complexity with problem size.
- Our current implementation of the mpQP solver is made in Matlab. One possible way of increasing the execution speed would be to implement the solver in some lower level language, like C or Fortran. However, as more than 80 percent of the execution time in the current implementation usually is spent on solving LPs (which already is implemented in Fortran), the faster execution speed obtained by implementing the solver in e.g. C would not be by orders of magnitude.
- Most of the LPs mentioned in the previous point are solved to remove redundant hyperplanes from representations of polyhedra. This method of removing redundant hyperplanes is easy to implement, but is sub-optimal with regard to execution speed. Thus, replacing this method with a more efficient one may be the most promising way of improving the mpQP solver in terms of execution speed.
- For instance, we consider using this kind of techniques for the automotive vehicle control problem in Chapter 4. This would separate this problem into a constrained control allocation part and a dynamical control problem. The controller can then command a yaw moment to the control allocation. We expect this to considerably decrease the complexity of the resulting control system implementation.

Bibliography

[1] V. Dua and E. N. Pistikopoulos, "An algorithm for the solution of multiparametric mixed integer linear programming problems," *Annals of operations research*, vol. 99, no. 1-4, pp. 123–139, 2000.

[2] Y. Wei, M. Fu, J. Ning, and X. Sun, "Quadratic programming thrust allocation and management for dynamic positioning ships," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 3, pp. 1632–1638, 2013.

[3] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[4] A. BEMPORAD, "A predictive controller with artificial lyapunov function for linear systems with input/state constraints," *Automatica*, vol. 34, no. 10, pp. 1255 – 1260, 1998.

[5] A. Bemporad, F. Borrelli, M. Morari *et al.*, "Model predictive control based on linear programming˜ the explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[6] 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.

[7] J. Mattingley, Y. Wang, and S. Boyd, "Code generation for receding horizon control," in *2010 IEEE International Symposium on Computer-Aided Control System Design (CACSD)*, 2010, pp. 985–992.

[8] Jacob, Y. Wang, and S. Boyd, "Receding horizon control," *Control Systems, IEEE*, vol. 31, no. 3, pp. 52–65, 2011.

[9] E. N. Pistikopoulos, V. Dua, N. A. Bozinis, A. Bemporad, and M. Morari, "On-line optimization via off-line parametric optimization tools," *Computers & Chemical Engineering*, vol. 26, no. 2, pp. 175–185, 2002.

[10] F. Borrelli, A. Bemporad, and M. Morari, "A Geometric Algorithm for Multi-Parametric Linear Programming," *Journal of Optimization Theory and Applications*, vol. 118, no. 3, pp. 515–540, Sep. 2003.

[11] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[12] A. Bemporad, F. Borrelli, and M. Morari, "Piecewise linear optimal controllers for hybrid systems," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 2, 2000, pp. 1190–1194 vol.2.

[13] J. Mattingley and S. Boyd, "Cvxgen: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[14] T. A. Johansen, T. I. Fossen, and S. P. Berge, "Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 1, pp. 211–216, 2004.

[15] T. A. Johansen, T. I. 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.

[16] A. J. Sørensen, "A survey of dynamic positioning control systems," *Annual Reviews in Control*, vol. 35, no. 1, pp. 123–136, 2011.

[17] T. A. Johansen and T. I. Fossen, "Control allocationa survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.

[18] M. Hanger, T. Johansen, G. Mykland, and A. Skullestad, "Dynamic model predictive control allocation using cvxgen," 2011, pp. 417–422.

[19] E. C. Kerrigan and D. Q. Mayne, "Optimal control of constrained, piecewise affine systems with bounded disturbances," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2. IEEE, 2002, pp. 1552–1557.

[20] F. Tan, H. Chen, M. Hou, and H. Zhao, "Model predictive trajectory tracking of a reentry hypersonic vehicle based on convex optimization," 2013, pp. 4167–4171, cited By 0.

[21] E. Pistikopoulos, A. Galindo, V. Dua, E. S. Kikkinides, L. Papageorgiou, W. Jorisch, K.-W. Benz, W. Neumann, M. Köhler, W. Fritzsche *et al.*, "Multi-parametric programming: Theory, algorithms and applications, volume," 2007.

[22] Y. Luo, A. Serrani, S. Yurkovich, M. W. Oppenheimer, and D. B. Doman, "Model-predictive dynamic control allocation scheme for reentry vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 1, pp. 100–113, 2007.

[23] T. I. Fossen and T. A. Johansen, "A survey of control allocation methods for ships and underwater vehicles," in *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*. IEEE, 2006, pp. 1–6.

[24] G. C. Goodwin, M. M. Seron, and J. A. De Doná, *Constrained control and estimation: an optimisation approach*. Springer Science & Business Media, 2006.