Jostein Vada

Prioritized Infeasibility Handling in Linear Model Predictive Control: Optimality and Efficiency

NTNU Trondheim
Norges teknisk-naturvitenskapelige universitet

Doktor ingeniøravhandling 2000:15
Institutt for teknisk kybernetikk

ITK-rapport 2000:1 W

Prioritized Infeasibility Handling in Linear Model Predictive Control: Optimality and Efficiency

Thesis by

Jostein Vada

Submitted in partial fulfillment of the requirements for the degree of Doktor Ingeniør

Department of Engineering Cybernetics
Norwegian University of Science and Technology
N-7491 Trondheim, Norway

February 2000

Report 2000-1-W

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of doktor ingeniør at the Norwegian University of Science and Technology (NTNU). The work has been carried out at the Department of Engineering Cybernetics, NTNU in the period September 1996 through December 1999, and was financed by The Research Council of Norway.

I am grateful to my supervisor, Professor Bjarne A. Foss, for accepting me as a doctoral student, and for his optimistic attitude, inspiration, and support. I would also like to thank my co-advisor, Associate Professor Tor Arne Johansen, for his enthusiasm, inspiration, and fruitful interaction. Moreover, I would like to thank my friend and office-mate during more than four years, dr. ing. Olav Slupphaug, for sharing his knowledge with me and for frequent and fruitful discussions. His sense of detail and interest in my work are also gratefully acknowledged.

Finally, I would like to thank my wife Kjersti and my four-year-old son Bendik. Kjersti for her love, support, and patience, and for being at home with Bendik during all the nights I have spent in my office, and giving me the opportunity to prioritize my research in a period when she was also a full-time student. Bendik for making me relax from papers and books when I was at home and for giving me a wider perspective on life.

Jostein Vada
Trondheim, February 2000

Summary

Model predictive control (MPC) relies on the successful solution of an optimization problem at each sample. In the optimization problem, the state of the controlled system comes in as a parameter on the right-hand side of the constraints along with possible time-varying constraint limits. Thus, due to modelling errors, disturbances, and operator intervention, one may encounter a situation where no solution exists for the corresponding optimization problem. In such situations the control input is not defined, and a relaxation of the constraints in the MPC optimization problem is desirable as an alternative to plant shutdown, operator intervention, etc.

The thesis contributes to research on optimal infeasibility handling in linear MPC when there is a prioritization to be made among the constraints, with focus on computational efficiency. Prioritization is a way to specify that some constraints are more important to satisfy than others. This prioritization can be used to determine the order in which the constraints should be relaxed when seeking an optimal relaxation which renders the corresponding optimization problem feasible. By optimal it is here meant that the violation of a lower prioritized constraint cannot be made less without increasing the violation of a higher prioritized constraint.

The first part of the thesis is devoted to optimal infeasibility handlers which are based on solving a sequence of optimization problems in order to compute an optimal constraint relaxation. Depending on the size of the optimization problem, computational capacity, and sampling time, optimal sequential infeasibility handlers may be too time consuming in order to be used in practise.

The main part of the thesis is concerned with the design of an infeasibility handler which, according to a given prioritization among the constraints, optimally relaxes an infeasible MPC optimization problem into a feasible one by solving a single-objective linear program (LP) on-line in addition to the standard on-line MPC optimization problem at each sample. By extending known results from the theory of preemptive multi-objective LP,

the existence of such a single-objective LP is proved, and by extending results from the theory of parametric LP, an efficient algorithm is developed for off-line computation of the parameters of this single-objective LP.

In the last part of the thesis, the proposed algorithm is applied to design an infeasibility handler for an MPC controlling the top part of a fluid catalytic cracker unit main fractionator. The results show that it is not intuitive to determine the parameters of the infeasibility handler. Stability of the proposed infeasibility handler combined with MPC is established. Further, some modifications of the proposed infeasibility handler are discussed such that it allows the same priority level to be shared among several constraints. These modifications can also be a useful strategy if the MPC problem at hand is so large that the computational load or the memory demands of the proposed off-line algorithm become prohibitive. Other strategies which reduce such problems are also discussed.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In all practical decision problems there will be constraints that limit the possible decisions. For instance, in order to decide when to leave Trondheim City by car to catch a plane which leaves Trondheim Airport at 5.00 p.m., and at the same time minimize the waiting time at the airport, one has to take the following constraints into consideration (among others): scheduled appointments on the day of departure, the latest check-in time, the speed and acceleration limit on the car, the amount of fuel in the car, the driving conditions, and the largest fine one is willing to pay. In addition, assume that one has present a model of the covered distance as a function of the speed. Also assume that one has an appointment scheduled from 3.00 p.m. to 4.15 p.m. on the day of departure, that the latest check-in time is 4.40 p.m, that the distance to drive is 35 km, and that one has to drive through the center of Trondheim on the way to the airport. Lastly, assume that one does not want to pay any fine. Given these constraints, there is no possibility to reach the airport and at the same time satisfy the given constraints, i.e. the decision problem is infeasible. The only way to remedy this situation is to modify (i.e. relax) one or more of the constraints. One possible relaxation is to cancel the appointment scheduled from 3.00 p.m. to 4.15 p.m., or to shorten the duration of the appointment. However, if it is very important not to change this appointment, one might after all be willing to pay a possible fine and thus modify this constraint, or even book a later plane. Thus, in this case, the relaxation one finally decides on is dependent on the relative importance of the different constraints. Later, it will become clear that this is the core of this thesis.

Model predictive control (MPC) is a mathematically defined decision problem, where one repetitively minimizes a cost function that is subject to a set of constraints. In MPC, the set of constraints consists of a model of the process to be controlled, physical limitations, constraints to ensure stability, and constraints which are related to desirables. The decision made by an MPC controller is a set of control inputs to the given process, and a new decision is recomputed at fixed instants (samples). If, at a given sample, there is no control input which satisfies the set of constraints, the optimization problem is infeasible and the corresponding control input is undefined. Such a situation is usually not acceptable, and thus, in order to make the MPC controller able to compute a control input, the constraints should be sufficiently relaxed in order to obtain a feasible optimization problem. Note that it is not always the case that there exists a relaxation which renders the corresponding optimization problem feasible. For instance, if there does not exist a decision that satisfies the set of physical constraints, there are no physically implementable relaxations which render the corresponding optimization problem feasible. Such a situation can be illustrated by using the above example: Assume that you are not allowed to change the ticket, that the appointment scheduled from 3.00 p.m. to 4.15 p.m. is so important that it cannot be changed, and that the maximum speed of the car is too low to reach the airport within 25 minutes. In such a situation, no constraint relaxation can make the decision problem feasible.

One might ask whether the original decision problem is badly designed if infeasibility problems occur. However, assume that the same cost function is to be optimized subject to the same set of constraints, for a set of different conditions. This is the case in MPC, where the state (condition) changes from one sample to the next. Moreover, due to modelling errors, unknown disturbances, and operator intervention, the state does not change exactly as predicted by the model inherent in the controller. Thus, during the design stage of the MPC problem, it is generally not trivial to forecast whether or not infeasibility problems will occur after an MPC controller is commissioned, i.e. put into operation in a real plant. Moreover, if, during the design stage of the MPC problem, the constraints were designed with the goal of avoiding infeasibility problems under any possible circumstances, one might have to avoid designing constraints which, under normal circumstances, would be natural to impose in order to obtain the desired performance and which do not cause infeasibility problems under normal circumstances. Thus, designing an MPC controller which does not, under any circumstances, run into infeasibility problems restricts the set of possible constraints that can be implemented in the MPC. This feature can be illustrated by returning to the transportation-to-the-airport problem: Assume that there are no appointments scheduled on the day of departure. Then there is no reason to violate the speed limits on the road, and thus in this case, the speed limits should be present as constraints in the decision problem. The presence of this constraint would probably also result in a more comfortable drive (desired performance). However, if the same constraints should be present under all conditions (i.e. regardless of any appointment), in order to guarantee feasibility under all conditions, the speed limit constraint on the road should be removed from the decision problem, or at least modified.

A procedure/mechanism designed to recover from infeasibility is denoted an infeasibility handler. The purpose of an infeasibility handler is, when possible, to compute sufficiently large relaxations of the constraints such that the modified optimization problem, obtained by replacing the original set of constraints with the modified set of constraints, becomes feasible. In order to obtain the best performance (in some manner), the computed relaxations should be minimized (in some manner). Note that, with a well designed infeasibility handler, one could design constraints which in advance we know will cause infeasibility problems.

In MPC, as in many other problems, it is often the case that some constraints are more important to satisfy than others. One way to explicitly express this difference in importance is to give the constraints different priorities. Imposing priority levels on the constraints is a systematic way to implement certain types of operational strategies of the type "it is more important to avoid emptying the separator, which may damage the downstream equipment, than to keep the pressure below a certain limit, since high pressure is handled by some kind of relief equipment". There are several ways to interpret the meaning of priority, and this interpretation must be reflected in the design of the corresponding infeasibility handler.

One strategy is to use a so-called soft constraint approach, where a term which penalizes a weighted norm of the constraint violations is added to the original cost function in the MPC optimization problem, where the weights in this term reflect the priority levels. There exists no systematic method for designing the weights in this penalty function, and in order to verify that the weights are suitable selected, a huge number of simulations are often required. In this approach, the priority levels are interpreted as soft priorities since the degree of importance of a given constraint is determined by the corresponding weight in the cost function.

As opposed to soft priorities, another strategy is to let the priority levels be interpreted as hard priorities, that is, interpret a constraint with a given priority level as being infinitely more important to satisfy than a lower prioritized constraint. Note that in the rest of this thesis, unless otherwise stated, prioritization means hard prioritization. Consequently, violation of a constraint with a given priority level is minimized regardless of the resulting violations of the lower prioritized constraints. This strategy allows for

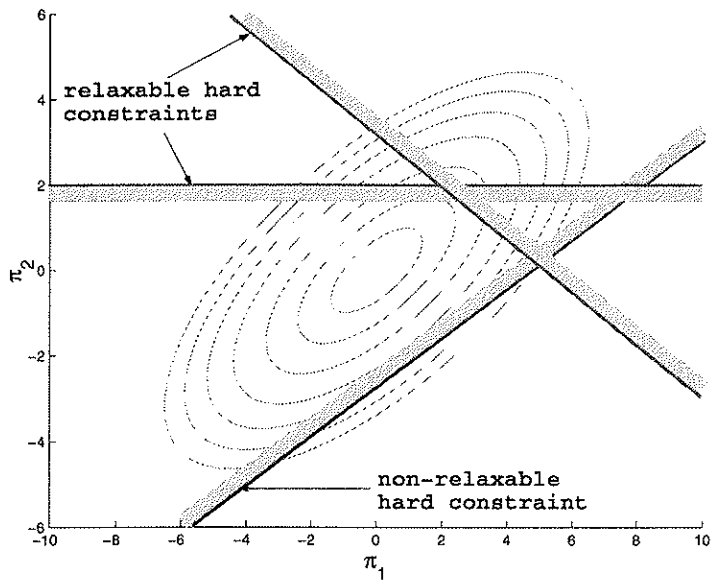

Figure 1.1: An infeasible optimization problem. There is no solution which satisfies all three constraints.

a systematic treatment of the constraints. By using hard prioritization, the relation between the specification and the achieved prioritization is explicit, and thus the design difficulties experienced by using soft prioritization are not present.

The effect of the use of hard constraints is illustrated by the optimization problem illustrated in Figure 1.1. In the figure, there are three constraints, two of them can be relaxed in order to obtain a feasible solution, and one cannot be relaxed. Each constraint is represented by a solid line, and the shaded area shows the half-plane where the corresponding constraint is infeasible. The ellipses represent contour lines of the cost function, and the minimum is located at $(0,0)$. It can be seen from the figure that the illustrated optimization problem is infeasible. Figure 1.2 shows where the optimum is located along with the corresponding violations in the case when both relaxable constraints are removed. Figure 1.3 shows the effect of using a hard prioritized approach in order to obtain a feasible optimization problem. The prioritization is shown in Figure 1.3. The relaxed constraint is illustrated by a solid line, and the corresponding original constraint is illustrated by a dotted line. Figure 1.4 shows the effect of exchanging the priority between the two relaxable constraints. It is observed that the opti-

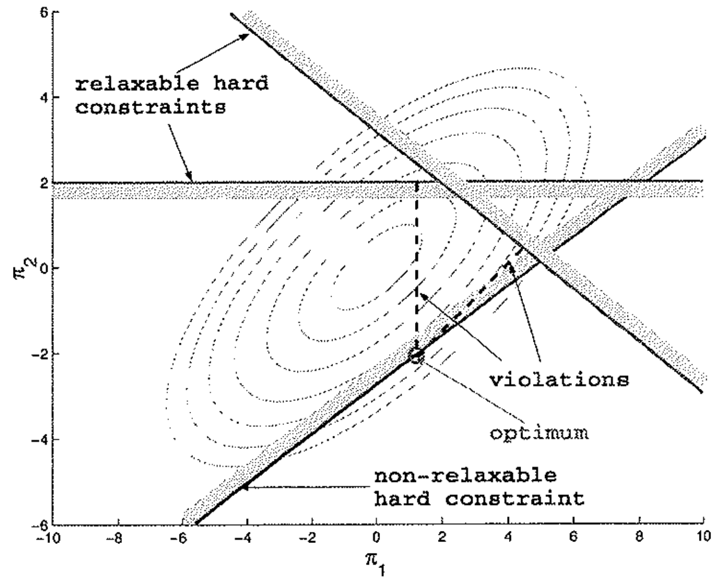

Figure 1.2: Effect of removing both relaxable constraints.

mum depends on the prioritization between the relaxable hard constraints. Note that in Figures 1.3 and 1.4, the optimum is uniquely determined by the constraints and the given prioritization. In other words the cost function has no influence on the optimum. Note that this is not generally the case.

1.2 Previous work

Linear MPC, i.e. MPC schemes where the prediction model of the plant is based on linear models, has been widely used in industry for more than 20 years, see e.g. (Garcia and Morshedi, 1986) and (Richalet, 1993) for industrial MPC applications. The main reason for its success is its ability to handle constraints on both the outputs and control inputs. There is a large number of theoretical contributions to this field, and in order to obtain a thorough insight into the theory of linear MPC, see e.g. (Lee, 1996), (Muske, 1995), (Zheng and Morari, 1995), (Rawlings and Muske, 1993), or (Muske and Rawlings, 1993), and in order to get an overview of industrial MPC, see e.g. (Qin and Badgwell, 1997). In this thesis, it is assumed that the reader knows the basics of linear MPC.

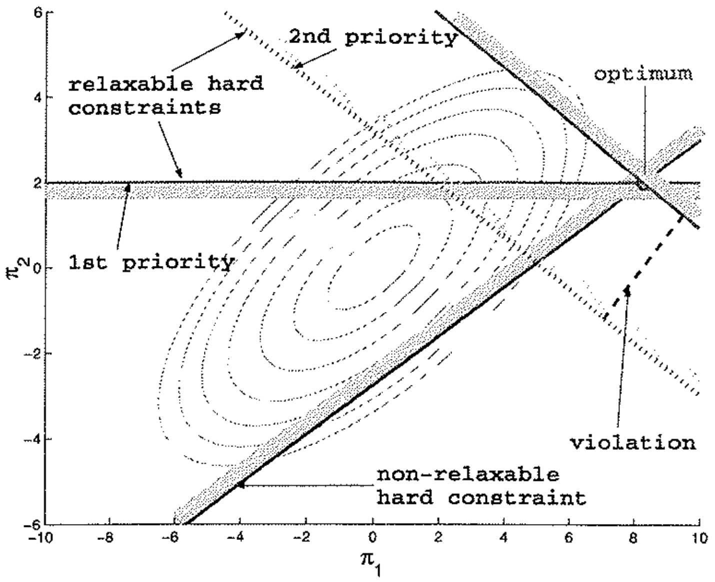

Figure 1.3: Effect of relaxation and prioritization. Note that only the lowest prioritized constraint is relaxed.

In the last few years, the research community has also turned attention to combining MPC with nonlinear models, see e.g. (Allgöwer, Badgwell, Qin, Rawlings and Wright, 1999). Note that MPC can also be used on plants with discrete and continuous control inputs and states as well, i.e. on hybrid systems, see e.g. (Bemporad and Morari, 1999) and (Slupphaug, Vada and Foss, 1997).

In this thesis, the focus is limited to infeasibility problems in linear MPC with continuous states and control inputs and a discrete-time model of the plant. A review of the contributions to this field is now presented. First, the approaches which do not use hard prioritization among the constraints are considered.

Defining some of the constraints as soft constraints will generally reduce the number of situations where an MPC controller runs into infeasibility problems compared to the case when all constraints are hard. One can conclude from the overview of industrial MPC given in (Qin and Badgwell, 1997) that most commercially available MPC technology includes soft constraints in their products. Setpoint approximation is another industrial strategy for handling constraints (Qin and Badgwell, 1997), where the idea is to predict future violations of the constraints, and then minimize the vio-

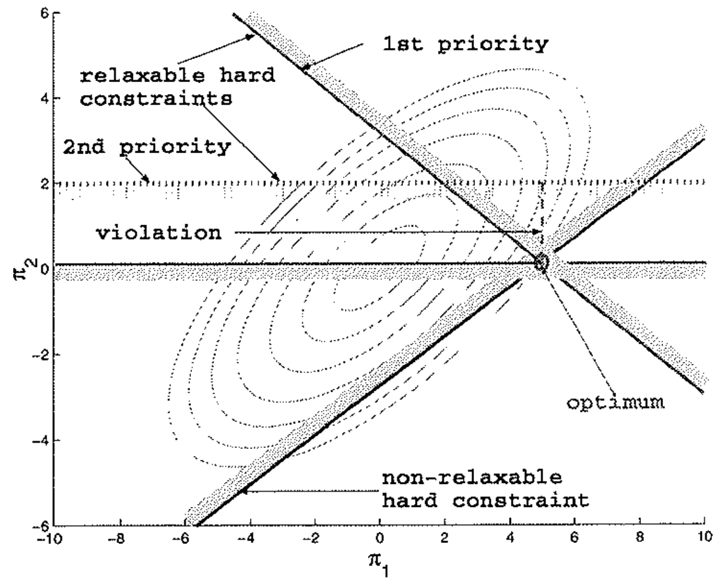

Figure 1.4: Effect of exchanging the prioritization of the two relaxable constraints in Figure 1.3.

lation of these constraints by introducing a setpoint that forces the output to stick to the boundary.

Oliveira and Biegler (1994) discuss stability of soft constrained MPC and investigates how the use of soft constraints affect the closed loop stability. Zheng and Morari (1995) show that with hard constraints on the control inputs and soft constraints on the outputs, linear MPC is globally asymptotic stabilizing if and only if the open loop system is stable. The use of soft constraints are also discussed in (Scokaert and Rawlings, 1999). They demonstrate that, by minimizing the square of the maximum violation over the horizon, as proposed in (Zheng and Morari, 1995), severe tuning difficulties caused by a mismatch in the open-loop and closed-loop predictions may occur. They also demonstrate that by minimizing the square of a weighted 2-norm of the constraint violations over the horizon, these problems do not occur.

In (Scokaert and Rawlings, 1999) the so-called optimal minimal time approach, which is an extension of an approach presented in (Rawlings and Muske, 1993), is presented. In this strategy, the minimal horizon, κ, is computed such that all constraints beyond κ can be satisfied on an infinite horizon. κ is computed by solving a sequence of optimization problems (one quadratic programming (QP) problem followed by a sequence of linear programming (LP) problems). Note that κ is dependent on the current state. At each sample, after computing κ, the constraint violations for the first κ samples are then minimized, and these violations are then used to relax the constraints in the original MPC formulations in order to obtain a feasible optimization problem. By using this strategy, minimizing the duration of constraint violations has higher priority than minimizing the size of the constraint violations.

Scokaert and Rawlings (1999) and Oliveira and Biegler (1994) also discuss the use of exact soft constraints, which is a soft constraint strategy which can be used in order to enforce fulfillment of the soft constraints whenever possible. In the exact soft constraint approach, as in ordinary soft constraint approaches, a term is added to the original cost function in the MPC optimization problem. However, instead of penalizing the square of the constraint violations in this term, a weighted l_1 norm of the constraint violations over the horizon is penalized. In order to obtain fulfillment of the soft constraints whenever possible, the weights in this term have to be large enough. Oliveira and Biegler (1994) show that MPC with exact soft constraints has the same stabilizing properties as ordinary hard constrained MPC.

In (Alvarez and de Prada, 1996), (Alvarez and de Prada, 1997), a heuristic infeasibility handler is proposed which treats the constraints on the control inputs and outputs in a separate manner. First, the input constraints are considered: If, after an operator intervention (e.g. due to a manual change of a limit or the value of a control input), the constraints on the control inputs cannot be satisfied (with the output constraints currently removed), the infeasibility handler either *i*) relaxes the values set by the operator, or *ii*) relaxes the constraints (within the physical limits), or uses a combination of *i*) and *ii*). For each control input, the user specifies which of these three methods is to be applied. If, after the relaxation of the constraints on the control inputs, the MPC optimization problem is still infeasible, the output constraints are the next candidates to be relaxed. The selections of which outputs to be relaxed are determined by inspecting the free (unconstrained) response of all outputs and then selecting the variables whose free responses violate their respective constraints as candidates for relaxation. In order to relax the output constraints, four strategies are proposed: *i*) modify the constraint horizon, *ii*) relax the value of the constraint, *iii*) relax the constraints on the control input, and *iv*) use a combination of *i*) to *iii*). For each of the outputs considered, one of the above strategies is to be selected. The violations of all output constraints using strategy *ii*) to *iv*) above are then computed by solving an LP problem, which minimizes the sum of all constraint violations over the constraint horizon. (Note that the idea of employing an LP in order to compute the constraint violation is also used in this thesis. However, the difference is that we compute the weights of this LP such that the resulting constraint violations are computed according to a given prioritization among the constraints.)

As several authors have emphasized (see e.g. (Garcia and Prett, 1986), (Meadowcroft, Stephanopoulos and Brosilow, 1992), (Qin and Badgwell, 1997), (Tyler and Morari, 1999)), it is often the case that some constraints are more important to satisfy than others, and designing a single objective function which reflects this difference in importance (as in the soft constraint approach) is difficult, if not impossible. Moreover, large weights are often needed in order to reflect the difference in importance, and if the difference in the weights becomes too large, numerical problems may occur when solving the corresponding optimization problem. Qin and Badgwell (1997) report a couple of commercial MPC vendors who include hard prioritized constraints in their software. In these applications, the following strategy is used: When the MPC optimization problem becomes infeasible, the lowest prioritized hard constraint is dropped, and the calculation is repeated. Since the violation of the lower prioritized constraints are not minimized in the presence of infeasibility, this strategy will generally result in unnecessarily large constraint violations. Moreover, a sequence of optimization problems needs to be solved, and thus in some applications it might be too time consuming.

Scokaert (1994) discusses several aspects of infeasibility handling, and he proposes a number of strategies to recover from infeasibility problems in cases when a prioritization is present among the constraints. One approach is to use the priority levels to determine a sufficiently large subset of the set of all constraints which, when discarded or relaxed, renders a feasible optimization problem. For the purpose of determining this set, he suggests two strategies. One strategy is to first compute the largest priority level i (large priority level means low priority) such that all constraints with priority level less than i can be satisfied. Then there is a minimization of the number of constraints with a priority level greater than or equal to i that needs to be violated in order to obtain a feasible optimization problem. The constraints in this set are then discarded. Another approach is to minimize the size of the set of constraints to be discarded, regardless of the priority levels, and if there are several candidates to this set, the set which maximizes the lowest priority level in this set is selected. These, and similar approaches can be very attractive in many MPC implementations, but the on-line computational load required will be prohibitively large in most cases. (Scokaert (1994) does not discuss how to compute these sets, however it is likely to believe that the solution of a large number of optimization problems is required.)

Meadowcroft et al. (1992) introduce the modular multivariable controller (MMC). The motivation behind their approach is that the controller design problem is in fact multi-objective, and that important design decisions cannot be elucidated if a single-objective optimization problem is designed in order to solve the original multi-objective problem. In their approach, setpoints and constraints are considered as objectives. For instance in soft constrained MPC, the design of the weights involves subjective judgments. Moreover, validation of whether the resulting controller meets the specified performance criteria or not is very hard to untangle. In the design of the MMC, the different objectives are given different priority levels, and a sequential strategy is used to compute the control input. However, this algorithm can be quite complicated for high dimensional systems with a high number of objectives (Tyler and Morari, 1999). Note that in (Meadowcroft et al., 1992) a detailed methodology is given just for the design of steady state MMCs.

Tyler and Morari (1999) present an approach for solving infeasibility in MPC problems with hard prioritized constraints. In their approach, logical (i.e. integer) variables are used to handle the priority ordering. By using their approach, only one mixed integer LP needs to be solved in order to compute the largest i such that each constraint with priority level 1 to i can be satisfied. However, in order to minimize the violation of the lower prioritized constraints according to the prioritization, a sequence of mixed integer LPs needs to be solved. Tyler and Morari (1999) compare the strategy they propose with a traditional strategy using a single cost function which penalizes a weighted sum of the squares of the constraint violations. With a given disturbance, they tuned the weights of the utility function such that the responses obtained by using the two approaches became equal. However, for another disturbance (equal in size), the traditional approach did not work well. They claim that choosing the weights such that the performance is satisfactory for both these disturbances is an unwieldy task.

Among the above cited works, only (Rawlings and Muske, 1993), (Scokaert, 1994), (Zheng and Morari, 1995), and (Scokaert and Rawlings, 1999) discuss the stability of the MPC controller combined with infeasibility handlers.

1.3 Contributions

This thesis contributes to research on infeasibility handling in linear model predictive control.

The main contribution in Chapter 2 is the discussion about how prioritization can be used among the constraints to specify how the constraints in an infeasible MPC optimization problem can be optimally relaxed in order to obtain a feasible optimization problem. A general algorithm for dealing with infeasibility problems is proposed. The algorithm is based on solving a sequence of optimization problems, and the algorithm allows for several strategies for minimizing the violations for each set of constraints having the same priority level. Compared to the commercial MPC schemes reported in (Qin and Badgwell, 1997) that use hard prioritized constraints, this approach generally reduces the constraint violations, since the commercial MPC schemes just drop the lower prioritized constraints in the case of infeasibility.

In Chapter 3, the problem of computing optimal constraint violations according to a given prioritization is formulated as a parametric preemptive multi-objective LP (parametric PMOLP). It is parametric, since in linear MPC, the right-hand side of the constraints is parameterized by the state and possibly by user-defined bounds on the constraints, and these parameters may change from one MPC optimization problem to another. It is preemptive due to the hard prioritization among the constraints. We prove the existence of a parametric single-objective LP such that for any parameter contained in a predefined set of parameters, any solution to this LP is also a solution to the parametric PMOLP. The key point is to select appropriate weights in the cost function of this single-objective LP. Further, we have developed an algorithm to compute these weights. Concerning linear MPC, a consequence of this result is that at each sample, in order to compute an optimal set of constraint violations according to a hard prioritization, only one LP needs to be solved on-line at each sample. To the best of my knowledge, all existing solution approaches to a parametric PMOLP rely on solving a sequence of optimization problems on-line.

In Chapter 4, the algorithm from Chapter 3 is applied in order to compute the parameters of the proposed infeasibility handler for a (simulated) realistic MPC problem. The results are promising and verify the practical viability of the algorithm. Moreover, it is shown that the proposed infeasibility handler combined with the Rawlings-Muske MPC-controller guarantees asymptotical stability of the origin, with a larger region of attraction than the Rawlings-Muske MPC-controller without an infeasibility handler. Finally, it is shown how to allow for several constraints to share the same priority level and how to combine the proposed infeasibility handler with a soft constrained approach.

Chapter 3 also contains extensions to some of the results in (Sherali, 1983) and (Gal, 1995): In (Sherali, 1983), it is proved that there exist weights to a single-objective LP problem (nonpreemptive problem) such that any optimum of this problem is optimal to a corresponding multi-objective LP problem with a prioritized ordering among the objectives (preemptive problem). In Chapter 3 this result is extended in that we consider problems where the right-hand side of the constraints is not fixed. In (Gal, 1995), single-objective LP problems with varying right-hand sides of the constraints are considered. In Chapter 3, some of the results in (Gal, 1995) on parametric programming are extended to consider preemptive problems as well.

1.4 Outline

First a few words on the structure of this thesis. Chapters 2, 3, and 4 are reprints of (Vada, Slupphaug and Foss, 1999), (Vada, Slupphaug and Johansen, 1999b), and (Vada, Slupphaug, Johansen and Foss, 2000) respectively, and thus there is some redundancy in these chapters, mainly in the introductions. Note that the papers occur as chapters in chronological order. Since the papers have been written over a period of almost two years, there might be some discrepancies in the choice of words and focus in these chapters. As an example, in Chapter 4 and Section 1.1, the notion of hard and soft prioritization is introduced, while in Chapters 2 and 3, the word prioritization is used to mean hard prioritization. The notation within each chapter is consistent, but there may be some minor inconsistencies in notation between the various chapters. Instead of providing the reader with a complete nomenclature list, the symbols are defined in the respective chapters.

The present chapter has presented the motivation for the present work, a literature review, and the main contributions in the thesis. The content of the following chapters is given below:

Chapter 2 formally presents the infeasibility problem in linear MPC and the use of priorities in order to obtain a clear specification of how to relax the constraints in order to obtain a feasible MPC optimization problem. A sequential solution approach is presented in order to compute the constraint relaxations, and the algorithm is illustrated by a simple example.

Chapter 3 presents an infeasibility handler which computes the optimal constraint relaxations according to a given prioritization by solving only one on-line LP at each sample. The existence of such an infeasibility handler is proved, and an algorithm is developed which computes the parameters to this infeasibility handler such that optimality according to the chosen prioritization is guaranteed.

Chapter 4 contains an application of the theory developed in Chapter 3, along with a stability result on the controller obtained by combining MPC with the proposed infeasibility handler. Some practical enhancements of the proposed infeasibility handler is also suggested in order to reduce the off-line computational complexity and simplify the engineering design of the MPC.

Chapter 5 ends the thesis and provides the main conclusions.

Chapter 2

A sequential approach to solve infeasibility problems

This chapter is a reprint of (Vada, Slupphaug and Foss, 1999), which was presented at the 14^{th} IFAC World Congress, Beijing, China, 1999.

Abstract All practical MPC implementations should have a means to recover from infeasibility. We propose an algorithm designed for linear state-space MPC which transforms an infeasible MPC optimization problem into a feasible one. The algorithm handles possible prioritizations among the constraints explicitly. Prioritized constraints can be seen as an intuitive and structural way to impose process knowledge and control objectives on the controlled process. The algorithm minimizes the constraint violations by solving a series of optimization problems, and the violation of a given constraint is minimized without affecting the higher prioritized constraints. An example shows the effect of implementing this algorithm on a simple process.

Keywords: Model based control, Constraints, Priority, Linear systems.

2.1 Introduction

During the last years, model predictive control (MPC) see e.g. (Rawlings, Meadows and Muske, 1994), has shown to become an attractive control strategy within the process industry. Important stability results within the area of linear MPC are given in (Rawlings and Muske, 1993). However, to fully exploit this stabilizing property, a means to recover from infeasibility of the associated optimization problem whenever possible is required, since generally, a practical MPC will sooner or later run into infeasibility problems. The infeasibility problems may e.g. be due to disturbances, operator intervention, or actuator failure. If the input constraints represent physical limitations (which is often the case) they must be enforced at all times. The state constraints are often desirables and should hence be satisfied whenever possible. Thus, usually, only the state constraints can be relaxed in order to transform the optimization problem into a feasible one in the case of infeasibility.

There exist techniques which transform an infeasible MPC-problem into a feasible one. Rawlings and Muske (1993) proposed to remove the constraints at the beginning of the horizon, i.e. for samples up to some sample number j_1. This feature is also implemented in the QDMC algorithm reported by Garcia and Morshedi (1986), who also proposed a soft constraint solution which minimizes the square of the constraint violations. The use of soft constraints is a way to *avoid* running into infeasibility problems. Zheng and Morari (1995) show that global asymptotic stability can be guaranteed for linear time-invariant discrete time systems with poles inside the closed unit disc subject to hard input constraints and soft output constraints. In (Scokaert and Rawlings, 1999) a method called optimal minimal time approach is proposed, which first minimizes the value of j_1, and then minimizes the size of the violation during the first j_1 samples of the prediction horizon.

Often, the state constraints are not equally important. One way to explicitly express this difference in importance is to give the state constraints different priorities. Imposing priority levels on the constraints is a systematic way to implement certain types of process knowledge, such as "avoiding shut-down is more important than discarding the product for some time, since start-up of the process is very expensive compared to discarding a certain amount of the product". The problem studied in this paper is how to allow control to be continued in the presence of infeasibility of the state constraints taking into account the information contained in the prioritization.

There are some existing techniques which take the prioritization levels into account when recovering from infeasibility. IDCOM-M from Setpoint Inc. provides a means for recovery from infeasibilities which involves prioritization of the constraints (Qin and Badgwell, 1997). When the calculation becomes infeasible, the lowest prioritized hard constraint is dropped, and the calculation is repeated. PCT from Profimatics also uses constraint prioritization when recovering from infeasibility (Qin and Badgwell, 1997). Meadowcroft et al. (1992) have developed a modular multivariable controller (MMC), which is based on the solution of a multiobjective optimization problem using the strategy of lexicographic goal programming where the objectives have different priorities. They proposed a detailed methodology for the design of steady state MMCs (they have left the detailed design of dynamic MMCs for a forthcoming publication).

In (Tyler and Morari, 1997) and (Tyler and Morari, 1999) it is presented an approach using integer variables for solving infeasible linear MPC problems where the constraints have different priorities. The minimization of the size of the violation of the constraints is done by solving a sequence of mixed integer optimization problems. They compare their methodology with conventional MPC by using weights and slack-variables on an example with 4 prioritized constraints. By trial and error, they find weights which give approximately the same performance for one specific disturbance. When a different disturbance enters their example process, the simulation results show that when using their approach, the two a highest prioritized constraints are fulfilled, while only the highest prioritized constraint is fulfilled when the conventional approach is used. This example supports the statement in (Qin and Badgwell, 1997), that for large problems it is not easy to translate control specifications into a consistent set of relative weights for a single objective function.

The main difference between our approach and the one presented in (Tyler and Morari, 1997) and (Tyler and Morari, 1999) (the so-called rigorous one) is that the latter approach results in a sequence of mixed integer LP (or mixed integer QP) problems in addition to the original MPC optimization problem, while our approach results in a sequence of LP (QP) problems in addition to the original MPC optimization problem. In both approaches, whether each step in the resulting sequence of optimization problems is an LP problem or QP problem depends on how the slackvariables associated with the constraints are penalized.

The outline of the paper is as follows: The next section presents the problem definition and the MPC formulation used. Then the algorithm which transfers an infeasible MPC optimization problem into a feasible one is presented, followed by a simulation example. The last section contains a discussion and some concluding remarks.

2.2 Problem definition

2.2.1 MPC formulation

The notation and MPC-formulation used here is adopted from (Scokaert and Rawlings, 1998). Consider the time-invariant, linear, discrete time system described by

$$x_{t+1} = Ax_t + Bu_t, \tag{2.1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ are the state and input vectors at discrete time t. It is assumed that (A, B) is stabilizable. The control objective is to regulate the state of the system optimally to the origin. The quadratic objective is defined over an infinite horizon and is given by

$$\phi(x_t, \pi) = \sum_{j=t}^{\infty} x_{j|t}^T Q x_{j|t} + u_{j|t}^T R u_{j|t}, \tag{2.2}$$

where $Q \geq 0$ and $R > 0$ are symmetric weighting matrices such that $(Q^{1/2}, A)$ is detectable, $\pi = \{u_{t|t}, u_{t+1|t}, \dots\}$, x^T is the transposed of x, and

$$x_{j+1|t} = Ax_{j|t} + Bu_{j|t}, \quad t \leq j \tag{2.3}$$

with $x_{t|t} = x_t$. The linear constraints are

$$\begin{aligned} Hx_{j|t} &\leq h, \; t < j \\ Du_{j|t} &\leq d, \; t \leq j \end{aligned}$$

where $h \in \mathbb{R}_+^{n_h}$ and $d \in \mathbb{R}_+^{n_d}$ ($\mathbb{R}_+$ is the positive reals) define the constraint levels, and H and D are the state and input constraint matrices respectively. The MPC optimization problem can now be defined as follows:

$$\begin{array}{lll} & \min_\pi \phi(x_t, \pi) & \\ \text{subject to:} & & \\ x_{t|t} & = x_t & \\ x_{j+1|t} & = Ax_{j|t} + Bu_{j|t}, & t \leq j \\ Hx_{j|t} & \leq h, & t < j \\ Du_{j|t} & \leq d, & t \leq j \\ u_{j|t} & = -Kx_{j|t}, & t + N \leq j \end{array} \tag{2.4}$$

The constraints need only be satisfied on a finite horizon to guarantee satisfaction on the infinite horizon (Rawlings and Muske, 1993). This form of MPC has Nm decision variables, and can be solved with standard quadratic programming solvers. K is discussed below. The performance index in the above equation can be formulated as (Rawlings and Muske, 1993)

$$\phi(x_t, \pi) = \sum_{j=0}^{N-1} (x_{j+t|t}^T Q x_{j+t|t} + u_{j+t|t}^T R u_{j+t|t}) + x_{j+N|t}^T \tilde{Q} x_{j+N|t}$$

where $\tilde{Q}$ is the solution of the matrix Lyapunov equation

$$\tilde{Q} = Q + K^T RK + (A - BK)^T \tilde{Q}(A - BK).$$

The feedback matrix K can be chosen in several ways. In the rest of this paper, $K = 0$ is used. In order to obtain stability, the unstable modi of the predictor, $x^u_{t+N|t}$, are zeroed at the Nth predicted sample (end point constraint), i.e.,

$$x^u_{t+N|t} = 0. \tag{2.5}$$

The feedback law is defined by receding horizon implementation of the optimal open-loop control. Given the optimal open-loop control strategy $\pi^*(x_t) = \{u^*_{t|t}(x_t), u^*_{t+1|t}(x_t), \dots\}$, the control law is thus given by

$$u_t(x_t) = u^*_{t|t}(x_t). \tag{2.6}$$

2.2.2 Compact problem formulation

Assume that the system, performance index, and predictor are given by (2.1), (2.2), and (2.3), respectively, and that the MPC problem formulation is given by (2.4) with $K = 0$ and the additional end point constraint (2.5). The problem studied in this paper is how to relax the state constraints in an optimal manner subject to their prioritization when the optimization problem defined by the MPC formulation becomes infeasible (e.g. due to a disturbance). The method solving this problem must be computationally implementable, and must not interfere with the control law defined by (2.4) and (2.6) when the optimization problem (2.4) is feasible.

2.3 Solution approach

2.3.1 The algorithm

When it is impossible to satisfy all state constraints simultaneously, it is desirable to satisfy as many of the highest prioritized constraints as possible. The violations of the other (infeasible) constraints should be minimized, taking their relative prioritization into account. The method described here is an *extension* of the theory presented in (Scokaert and Rawlings, 1999), such that the constraint violations are minimized according to their priorities. Operating at Pareto-optimal points in the "size of violation - duration of violation" space is the goal. The MPC problem defined in (2.4) (with $K = 0$), can be rewritten as[1]

$$\min_\pi \phi(x_t, \pi) \quad \text{subject to:} \tag{2.7}$$

$$\text{"hard" hard constraints} \begin{cases} x_{t|t} & = x_t \\ x^u_{t+N|t} & = 0 \\ x_{j+1|t} & = Ax_{j|t} + Bu_{j|t}, \; t \leq j \\ Du_{j|t} & \leq h, \qquad t \leq j \\ u_{j|t} & = 0, \qquad t+N \leq j \end{cases} \tag{2.8}$$

$$\text{"soft" hard constraints} \begin{cases} c_1 : \; H^1 x_{j|t} \leq h^1, \; t < j \\ \vdots \\ c_{n_c} : H^{n_c} x_{j|t} \leq h^{n_c}, \; t < j \end{cases} \tag{2.9}$$

where the constraints marked as "hard" hard constraints cannot under any circumstances be violated, since they are either physical limitations on the process, or related to zeroing the unstable modi at the end of the prediction horizon, or decided by the move horizon N which is assumed to be fixed. The constraint sets $\{c_1, \ldots, c_{n_c}\}$ are constructed such that constraint set c_i has higher priority than c_{i+1}. A constraint set is composed of one or more scalar constraints having the same priority. $H_i \in \mathbb{R}^{n_{c_i} \times n}$ and $h_i \in \mathbb{R}_+^{n_{c_i}}$, where n_{c_i} is the number of constraints in constraint set c_i. An algorithm solving the problem of infeasibility subject to the prioritization among the constraints is presented next. In the algorithm, a sequence defined as $\{c_l, \ldots, c_m\}$, $l > m$, is interpreted as the empty set.

Step 1: Solve the optimization problem defined by (2.7), (2.8) and (2.9). If a feasible solution exists, the optimal solution (π^*) is found - terminate. Else, the problem infeasible. Go to step 2.

Step 2: Check existence of a solution to (2.8). If there does not exist any solution, the process cannot be stabilized with the given controller. Some kind of extraordinary action has to be taken. Else, if there exist a solution, set $k \leftarrow 1$ and go to Step 3. Note that the integer k is indexing the constraints, and is *not* related to time.

Step 3 Check existence of a solution to (2.8) and (2.9), but without constraint sets $\{c_{k+1}, \ldots, c_{n_c}\}$. Go to Step 4.

Step 4 If a feasible solution is found, set $k \leftarrow k+1$, and go to Step 3. Else, if no feasible solution is found, constraint set $\{c_1, \ldots, c_k\}$ cannot be satisfied simultaneously. Go to Step 5.

[1]Detectability of $(Q^{1/2}, A)$, which is a general requirement in Section 2.2.1, is not nescessary for stability here because of the end point constraint $x^u_{t+N|t} = 0$.

Step 5 Step 4 showed that constraint set c_k cannot be satisfied when $\{c_1, \dots, c_{k-1}\}$ are satisfied. Minimize the violation of constraint set c_k, i.e. compute optimal slack variables $(\Delta h^k_{j|t})^* \in \mathbb{R}^{n_{c_k}}_+$, such that

$$c'_k\colon H^k x_{j|t} \leq h^k + (\Delta h^k_{j|t})^*, \quad t < j \tag{2.10}$$

and $\{c_1, \dots, c_{k-1}\}$ are satisfied. There are several ways to compute the optimal slack variables, according to the control policy, see the discussion at the end of this section. Set $n_s \leftarrow 1$, where n_s is number of softened "soft" hard constraints. If $k < n_c$, i.e. there are more slack variables to be computed, go to Step 6, else go to Step 8.

Step 6: Minimize the violation of constraint set c_{k+n_s}, i.e. compute optimal slack variables $(\Delta h^{k+n_s}_{j|t})^*$, using the same strategy as in Step 5, such that $\{c_1, \dots, c_{k-1}, c'_k, \dots, c'_{k+n_s-1}\}$ are satisfied. Go to Step 7.

Step 7 If $k + n_s < n_c$, i.e. there are more slack variables to be computed, set $n_s \leftarrow n_s + 1$ and go to Step 6, else go to Step 8.

Step 8 At this step, the status is as follows: Constraint sets $\{c_1, \dots, c_{k-1}\}$ are not violated, and $\{c_k, \dots, c_{n_c}\}$ are replaced by $\{c'_k, \dots, c'_{n_c}\}$ such that there exist a solution which fulfills $\{c_1, \dots, c_{k-1}, c'_k, \dots, c'_{n_c}\}$. Now, with the last degrees of freedom (if any), minimize the performance index (2.7) subject to these constraints.

As stated in Step 5 above, there are several ways to compute the optimal slack variables for a given constraint set. In Section 4 below, the optimal minimal time approach (Scokaert and Rawlings, 1999) is used to compute the slack variables within each constraint set. Considering constraint set c_k, the computation can be described as follows: Let $\kappa^k(x_t)$ denote the least integer such that c_k can be fulfilled when $j \geq t + \kappa^k(x_t)$. Given $\kappa^k(x_t)$, the least (in some sense) $\Delta h^k_t \in \mathbb{R}^{n_{c_k}}_+$, i.e. $(\Delta h^k_t)^*$, is computed such that c'_k defined in (2.10) is feasible if $(\Delta h^k_{j|t})^* = (\Delta h^k_t)^*$ when $t < j < t + \kappa^k(x_t)$ and $(\Delta h^k_{j|t})^* = 0$ when $t + \kappa^k(x_t) \leq j$. Another method is, for each constraint set, to introduce different priorities for every sample. If, for example, the constraints at predicted sample q in constraint set k has higher priority than the constraints at predicted sample $q - 1$ in constraint set k, then the minimal duration of the constraint violation is obtained, but the size of the violations will generally differ from the corresponding size of violations resulting from the optimal minimal time approach. Another method for computing $(\Delta h^k_{j|t})^*$ is to minimize $\sum_{j=t+1}^{P} (\Delta h^k_{j|t})^T W_j \Delta h^k_{j|t}$ subject to c_1 to c_{k-1} and $H^k x_{j|t} \leq h^k + \Delta h^k_{j|t}$, $t < j \leq t + P$, where $W_j \in \mathbb{R}^{n_{c_k} \times n_{c_k}}$ is a weighting matrix, and P is sufficiently large such that $\Delta h^k_{j|t} = 0$, $j > t + P$ is feasible.

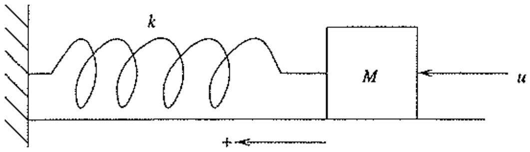

Figure 2.1: An idealized mass-spring system.

2.3.2 Stability

The controller proposed in this paper is based on the controller given in (Rawlings and Muske, 1993), which is shown to be exponentially stabilizing in (Scokaert and Rawlings, 1998), and is essentially an add-on of infeasibility handling in an optimal manner taking priorities into account. This add-on does not interfere with any of the stabilizing properties of the controller given in (Rawlings and Muske, 1993), hence stability is retained.

2.4 Example

2.4.1 Process

The proposed method described above will now be implemented on a simple example, and compared, insofar it is possible, with the optimal minimal time approach described in (Scokaert and Rawlings, 1999). The process used in this example is an idealized mass-spring system which is illustrated in Figure 2.1. The spring is assumed to be linear, and the mass slides without any friction. There is a force u directed horizontally on the mass. It is assumed that both the position and the velocity of the mass are ideally measured, and that the spring constant k and the mass M both are equal to 1.0. By using exact discretization with sample time equal to 0.5 s, the system is given by the following equations:

$$x_{t+1} = Ax_t + Bu_t,$$

where

$$A = \begin{bmatrix} 0.8776 & 0.4794 \\ -0.4794 & 0.8776 \end{bmatrix}, \quad B = \begin{bmatrix} 0.1224 \\ 0.4794 \end{bmatrix}.$$

The MPC problem is given by (2.4) and (2.2) with $K = 0$, $Q = I$, $R = 1$, $N = 5$ and

$$H = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \; h = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}, \; D = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \; d = \begin{bmatrix} 0.6 \\ 0.6 \end{bmatrix}.$$

The predictor is equal to the process. The input constraint has always to be satisfied, and the prioritization of the state constraints is as follows, in descending order:

i. Constraints on the mass position, i.e. $|x_{j|t,1}| \leq 0.25, \; j > t$.

ii. Constraints on the mass velocity, $|x_{j|t,2}| \leq 0.25, \; j > t$.

Using the notation in Section 2.3.1, the following constraint sets are defined:

$$\begin{aligned} c_1 &: H^1 x_{j|t} \leq h^1, \; j > t, \\ c_2 &: H^2 x_{j|t} \leq h^2, \; j > t \end{aligned}$$

where

$$H^1 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}, \; H^2 = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}, \; h^1 = h^2 = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}.$$

2.4.2 Simulation: Case 1 - applying the proposed approach

At time $t = 0$, a state disturbance of $[0.8, -0.4]^T$ enters the system and the approach described in Section 2.3.1 is used to recover from infeasibility. Step 1 to 4 in the algorithm show that only the "hard" hard constraints can be satisfied at time $t = 0$. In Step 5, the optimal minimal time approach described in (Scokaert and Rawlings, 1999) is used to compute the minimal violation of constraint set c_1: First the minimal time, $\kappa^1(x_t)$, beyond which the constraints c_1 can be satisfied is computed. Given $\kappa^1(x_t)$, the following LP is solved to compute the minimal size of the constraint violation of constraint set c_1:

$$\begin{aligned} &\min_{\pi_t, \Delta h_t^1} S\Delta h_t^1, \text{ subject to} \\ &\text{"hard" hard constraints} \\ &H^1 x_{j|t} \leq h^1 + \Delta h_t^1, \quad t < j < t + \kappa^1(x_t) \\ &H^1 x_{j|t} \leq h^1, \quad t + \kappa^1(x_t) \leq j \end{aligned} \tag{2.11}$$

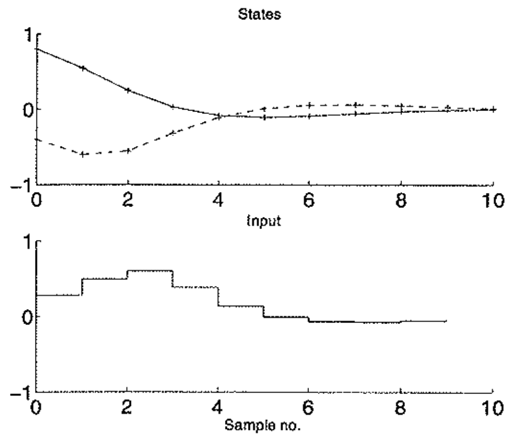

Figure 2.2: The figure shows the simulation results from Case1. In the upper part, the solid line shows $x_{1,t}$ and the dashed line shows $x_{2,t}$. Dotted lines are the constraint limits.

where $S = [1\ 1]$. At time $t = 0$, the optimal slack variables computed by this LP, are $(\Delta h_0^1)^* = [0.2944, 0.0]^T$, and $\kappa^1(x_0) = 3$. Let c_1' denote the two last constraints in (2.11) when $\Delta h_t^1 = (\Delta h_t^1)^*$. In Step 6, the minimal violation of constraint set c_2 is computed by first computing the minimal time, $\kappa^2(x_t)$, beyond which the constraints c_2 can be satisfied, subject to the "hard" hard constraints and c_1'. Given $\kappa^2(x_t)$, the minimal size of the constraint violation of constraint set c_2 is computed by a LP problem similar to (2.11), but with c_1' added to the hard constraints. The optimal slack variables computed by this LP at $t = 0$ are $(\Delta h_0^2)^* = [0.3509, 0.0]^T$, and $\kappa^2(x_0) = 4$. Let c_2' denote the relaxed constraint set c_2. In Step 8, the performance index (2.7) is minimized subject to the "hard" hard constraints, c_1' and c_2'. The receding horizon implementation using this strategy on the example process with the given disturbance results in the response shown in Figure 2.2. It can be observed from the figure that the constraints in constraint set c_1 are satisfied when $j \geq 2$, while $\kappa^1(x_0) = 3$. This difference in open- and closed-loop is due to the receding horizon nature of MPC and finite move horizon (N).

2.4.3 Simulation: Case 2- the optimal minimal time approach

The same disturbance as in Case 1 enters the system at time $t = 0$, but now all slack variables are minimized upon simultaneously by using the optimal minimal time approach presented in (Scokaert and Rawlings, 1999),

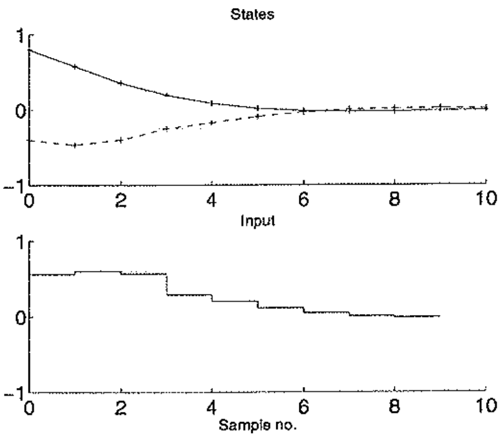

Figure 2.3: The figure shows the simulation results from Case 2. In the upper part, the solid line shows $x_{1,t}$ and the dashed line shows $x_{2,t}$. Dotted lines are the constraint limits.

	# LP problems	# QP problems
Case 1:	16	22
Case 2:	13	16

Table 2.1: Computational load for the simulation examples Case 1 and Case 2

i.e. there are no prioritization among the constraints. This approach is equal to the approach used in the previous section, but with all constraints collected in constraint set c_1, and gives the response shown in Figure 2.3. It can be seen that the violations of constraint set c_1 are less, both in time and size, when the approach used in this paper is applied, compared to the plain optimal minimal time approach. The expense is larger violations of constraint set c_2, this is of course due to the prioritization. It can be observed that in this case, the approach described in this work causes a longer time period with constraint violations, compared to the plain optimal minimal time approach.Table 2.4.3 shows the number of LP and QP problems needed to be solved in Case 1 and 2. It can be seen that, for the example presented here, the total number of LP and QP problems generated by the approach presented in this paper is about 30% greater than the optimal minimal time approach presented in (Scokaert and Rawlings, 1999). It should be noted that the number of constraints in the QP and LP problems generated by the two approaches are different. At a given sample, the number of constraints in the LP problems generated by our approach are less than or equal to the number of constraints generated by the optimal minimal time approach, since only a subset of the state constraints is present when computing κ^k in our priority handling approach, while all state constraints are present when computing κ in the optimal minimal time approach.

2.5 Discussion/Conclusion

All practical MPC implementations should have a means to recover from infeasibility, and this paper contains an algorithm which transforms an infeasible hard constrained MPC optimization problem into a feasible one by relaxing those hard state constraints which do not affect the stability of the controlled process. This is done taking into account priorities among the state constraints and extending the work (Scokaert and Rawlings, 1999), where priorities are not handled. The assignment of priorities is an intuitive and natural means to state certain objectives on the controlled process, and as shown in this paper, MPC is a suitable framework to impose such objectives. When minimizing the violation of a given constraint, the violations of the higher prioritized constraints are not affected.

If some of the input constraint are desirables rather than physical constraints, the infeasibility handling algorithm should also take these constraints into account. Extending the proposed algorithm to include input constraints is trivial.

The example shows how the violations of a constraint are minimized upon at the expense of larger violations of the lower prioritized constraint. In the example, the optimal minimal time approach (Scokaert and Rawlings, 1999) is used to minimize the constraint violations of the constraints which have the same priority. Other approaches may also be used. Consider, for example, a process where large sizes of constraint violations are very expensive. In some cases (such as in non-minimum-phase processes) it will then often be more cost efficient to allow for longer duration of violations at the benefit of smaller sizes of violations. Anyway, as mentioned earlier, obtaining Pareto optimal operation in "the space of duration of violation and size of violation" should always be the goal.

In the example, the number of optimization problems needed to be solved when our approach is used, is about 30% larger than the number of optimization problems needed to be solved when the optimal minimal time approach is used. In other examples, the difference in computational load will probably be different. Whether the computational load required by our approach is acceptable in practise is dependent on the process dynamics and the sampling period of the given process in addition to the computational capacity of the computer where the MPC is installed.

Chapter 3

A Parametric Preemptive Multi-Objective Linear Programming Approach

This chapter, except for Section 3.4.4, Section 3.C, and Section 3.D, is a reprint of (Vada, Slupphaug and Johansen, 1999b). (Vada, Slupphaug and Johansen, 1999b) was submitted to Journal of Optimization Theory and Applications in October 1999. Parts of the results in this are also given in (Vada, Slupphaug and Johansen, 1999a).

Abstract All practical model based control (MPC) implementations should have a means to recover from infeasibility. We propose a strategy designed for linear state-space MPC with prioritized constraints. It optimally relaxes an infeasible MPC optimization problem into a feasible one by solving a single-objective linear program (LP) on-line in addition to the standard on-line MPC optimization problem at each sample. By optimal it is meant that the violation of a lower prioritized constraint cannot be made less without increasing the violation of a higher prioritized constraint. The problem of computing optimal constraint violations is naturally formulated as a parametric preemptive multi-objective LP. By extending well known results from parametric LP, the preemptive multi-objective LP is reformulated into an equivalent standard single-objective LP. An efficient algorithm for off-line design of this LP is given, and the algorithm is illustrated on an example.

Keywords: Parametric programming, Preemptive programming, Infeasibility, Linear model predictive control.

3.1 Introduction

Model predictive control (MPC) (or, receding horizon control or moving horizon control) (Muske and Rawlings, 1993), (Rawlings and Muske, 1993) has become an attractive control strategy within the process industry. In MPC, at each time step, an sequence of optimal predicted control inputs is found by solving a constrained optimal control problem on some horizon into the future. The first element of the optimal predicted input sequence is used as present control input, and the same procedure is repeated at the next time step. In practical MPC implementations, a means to recover from infeasibility[1] of the associated optimization problem whenever possible is required. Typically, some of the constraints, such as physical limitations, must be enforced at all times, while other constraints can be relaxed in order to transform the optimization problem into a feasible one in the case of infeasibility.

There exist techniques which transform an infeasible MPC-problem into a feasible one by treating equally all constraints which can be relaxed, see e.g. (Garcia and Morshedi, 1986), (Rawlings and Muske, 1993), (Qin and Badgwell, 1997) and (Scokaert and Rawlings, 1999). However, the constraints are often not equally important, e.g. a safety constraint is usually more important to satisfy than a product quality constraint. One way to explicitly express this difference in importance is to give the constraints different priorities. Imposing priority levels on the constraints is a systematic way to implement certain types of operational strategies of the type "it is more important to avoid emptying the separator, which may damage the downstream equipment, than to keep the pressure below a certain limit, since high pressure is handled by some kind of relief equipment". There are some existing techniques which take such prioritization levels into account when recovering from infeasibility. IDCOM-M (Setpoint Inc.), HIECON and PFC (both from Adersa) provide a means of recovering from infeasibilities which involves prioritization of the constraints. When the on-line optimization problem becomes infeasible, the lowest prioritized constraints are dropped (Qin and Badgwell, 1997).

Scokaert (1994) discusses issues related to the problems of infeasibility in constrained predictive control, and proposes several strategies to solve such problems, including strategies involving priority levels. The most rigorous approach he proposes for infeasibility handling is to satisfy as many of the highest prioritized constraints as possible, and then compute a feasible relaxation of the other constraints by treating them as soft constraints, that

[1] An optimization problem is said to be *infeasible* or *inconsistent* if the constraints cannot be satisfied.

is, a term is added to the cost function in the original MPC optimization problem which penalizes the violations of these constraints. However, he does not discuss *how* to compute the set of constraints which can be satisfied without any violation.

In (Tyler and Morari, 1999) an approach is presented for solving infeasible linear MPC problems where the constraints have different priorities. In that approach, integer variables are introduced to cope optimally with the prioritization. The minimization of the size of the violation of the constraints is performed according to their prioritization by solving a sequence of mixed integer optimization problems.

In (Vada, Slupphaug and Foss, 1999) an algorithm is presented which, in case of infeasibility of the MPC-problem, optimally takes the prioritization among the constraints into account when relaxing the constraints. This algorithm includes a sequence of LP or QP problems to be solved at every sample. The main difference between the approach described in (Vada, Slupphaug and Foss, 1999) and the one presented in (Tyler and Morari, 1999) is that the latter approach results in a sequence of mixed integer LP (or mixed integer QP) problems in addition to the original MPC optimization problem, while the former approach results in a sequence of LP (or QP) problems in addition to the original MPC optimization problem. Note that the number of optimization problems needed to be solved in the first approach is generally less than in the latter approach. However, if the sampling time is short compared to the number and size of the optimization problems to be solved, both approaches may be prohibitively time consuming.

An important difference between the algorithms presented in (Tyler and Morari, 1999), (Scokaert, 1994), and (Vada, Slupphaug and Foss, 1999) and the other approaches mentioned above which *also* take prioritization into account, is that the algorithms in (Scokaert, 1994), (Vada, Slupphaug and Foss, 1999) and (Tyler and Morari, 1999) minimize the violations of those constraints which cannot be fulfilled. Just dropping a set of constraints may result in unnecessary large constraint violations, and will in this sense be suboptimal.

In (Meadowcroft et al., 1992) a modular multivariable controller (MMC) is developed, which is based on the solution of a multi-objective optimization problem using the strategy of lexicographic goal programming where the objectives have different priorities. This solution strategy implies that the optimization problem is solved sequentially, and thus suffers from the same problems related to computational time as the approaches in (Tyler and Morari, 1999) and (Vada, Slupphaug and Foss, 1999). ((Meadowcroft et al.,

1992) contains a detailed methodology for the design of steady state MMCs only.)

The main contribution of this paper is that it is shown that the algorithm presented in (Vada, Slupphaug and Foss, 1999), in the case when all constraints have different priorities, can be reduced from a sequence of LP problems to a single LP problem by properly selecting the weights (or, cost vector) in this LP problem. These weights are computed *off-line*. The main idea is to select sufficiently large weights on the higher prioritized constraints relative to the lower prioritized constraints in order to optimally satisfy the given prioritization. A consequence of this is a great reduction of the on-line computational demands of the infeasibility handler. Existence of such weights is proved under non-restrictive assumptions, and it is shown how these weights can be computed in order to avoid unnecessarily large weights.

This work is related to (Sherali, 1983) and (Gal, 1995). In (Sherali, 1983), it is proved that there exist weights to a single-objective LP problem (non-preemptive problem) such that any optimum of this problem is optimal for a corresponding multi-objective LP problem with a prioritized ordering among the objectives (preemptive problem). The present paper extends this result in that we consider problems where the right hand side of the constraints is not fixed. This is motivated by the fact that in the MPC setting, the right hand side of the constraints depends on both the current state and constraint limits. In (Gal, 1995), single-objective LP problems with varying right hand sides of the constraints are considered. In the present paper, some of the results in (Gal, 1995) on parametric programming are extended to consider preemptive problems as well.

The existence results contained herein are based on the conference paper (Vada, Slupphaug and Johansen, 1999a), while the algorithm presented for solving the weight design problem has not been published elsewhere.

The outline of the paper is as follows: In the next section, the linear MPC optimization problem with infeasibility handling is stated, followed by a definition of the weight design problem. Then, in Section 3.3, the existence of a solution to this problem is established, and in Section 3.4 we present an algorithm which solves it. Finally, the theory is illustrated by an example in Section 3.5.

The following notation is used throughout the paper: Let $x, y \in \mathbb{R}^n$. $x \geq (>) y \Leftrightarrow x_i \geq (>) y_i$, $i = 1, \ldots, n$. (x, y) is used to express $[x^T, y^T]^T$. $0_{n \times m}$ is a matrix of dimension $n \times m$ with zeros, 0_m is an m-dimensional vector with zeros, and I_n is the $n \times n$ identity matrix. $A_{i,j}$ denotes the jth element in the ith row of A. e_k is the kth unit vector. $\mathbb{I}_n := \{0, \ldots, n\}$, and $\mathbb{I}_n^+ := \{1, \ldots, n\}$, where $n \geq 1$ is an integer. $|J|$ denotes the cardinality of the set J.

3.2 Problem definition

In Section 3.2.1, the constraints associated with prioritized MPC are formulated. These constraints are based on the MPC-algorithm of (Rawlings and Muske, 1993). Next, the algorithm presented in (Vada, Slupphaug and Foss, 1999) is stated for the case when all constraints have different priorities. This algorithm requires the solution of a sequence of LPs. In Section 3.2.2, a problem, whose solution can be used to reduce this sequence of LP problems into a single LP problem, is stated.

3.2.1 Prioritized constraints in linear MPC

Let the system to be controlled be described by

$$x_{t+1} = f(x_t, u_t, \eta_t), \tag{3.1}$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, and $\eta_t \in \mathbb{R}^{n_\eta}$ denote the state-, control- and disturbance vector at time t, respectively. A linear predictor is given by

$$x_{j+1|t} = Ax_{j|t} + Bu_{j|t}, \; j \geq t, \tag{3.2}$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $x_{j|t} \in \mathbb{R}^n$ and $u_{j|t} \in \mathbb{R}^m$ are the predicted state and input vector at future time j, respectively. Note that the system (3.1) and predictor (3.2) are equal if $f(x, u, \eta) = Ax + Bu$. This defines the nominal case. The reason we let the "real" system be described by a nonlinear disturbed system is to emphasize the fact that this will always be the case in reality, and is one of the reasons why infeasibility handling is needed. However, stability results are typically stated for the nominal case. In this paper we do not consider stability analysis. The input constraints are of the form

$$\begin{array}{rcl} Du_{j|t} & \leq & d > 0, \; t+N > j \geq t \\ u_{j|t} & = & 0, \qquad j \geq t+N, \end{array} \tag{3.3}$$

where $D \in \mathbb{R}^{n_d \times m}$, $d \in \mathbb{R}^{n_d}$, and N is the move horizon. Similarly, the state constraints are of the form

$$Hx_{j|t} \leq h > 0, \; j = t+1, \ldots \tag{3.4}$$

where $H \in \mathbb{R}^{n_h \times n}$ and $h \in \mathbb{R}^{n_h}$. Note that the constraints are defined on an infinite horizon. In (Rawlings and Muske, 1993) it is shown that for a given (bounded) initial state $x_{t|t}$, if (A, B) is stabilizable and N is large enough, there exists a (finite) integer $j_2 \geq N$ such that satisfaction of $Hx_{j|t} \leq h$ up to and including time $t + j_2$ guarantees satisfaction of the state constraints on the infinite horizon. Thus, given such a j_2, in the nominal case the total number of scalar state constraints in the MPC optimization problem (to be presented at the end of this section) is $n_h j_2$.

If the predictor is unstable, the unstable modi of the predictor are zeroed at the end of the move horizon for stability reasons (Rawlings and Muske, 1993). This stability constraint can be expressed as

$$\Gamma U = \gamma x_{t|t}, \tag{3.5}$$

where $U := (u_{t|t}, u_{t+1|t}, \cdots, u_{t+N-1|t})$, $\Gamma := \tilde{V}_u[A^{N-1}B, A^{N-2}B, \cdots, B]$, $\gamma := -\tilde{V}_u A^N$, where $\tilde{V}_u$ is defined by partitioning the Jordan form of the A matrix into stable and unstable parts

$$A =: VJV^{-1} =: [V_u \; V_s] \begin{bmatrix} J_u & 0 \\ 0 & J_s \end{bmatrix} \begin{bmatrix} \tilde{V}_u \\ \tilde{V}_s \end{bmatrix},$$

in which the unstable eigenvalues of A and the eigenvalues of J_u are equal (Muske and Rawlings, 1993). The open-loop optimization problem (MPC problem) is to minimize the following quadratic objective function on an infinite horizon

$$\min_U \sum_{k=0}^{\infty} (x_{k+t|t}^T Q x_{k+t|t} + u_{k+t|t} R u_{k+t|t})$$

subject to $x_{t|t} = x_t$, (3.2), (3.3), (3.4) and (3.5), where $Q \in \mathbb{R}^{n \times n}$ is positive semidefinite and $R \in \mathbb{R}^{m \times m}$ is positive definite. Only the first move is applied to the plant (3.1), i.e. $u_t = u_{t|t}$, and at the next sample, the procedure is repeated. In the nominal case, this control law is exponentially stabilizing if and only if the corresponding optimization problem is feasible (follows from (Scokaert and Rawlings, 1999)). However, for certain states x_t, this optimization problem may be infeasible (e.g. due to model errors, disturbances, operator intervention or "narrow" constraints), and this situation motivates the problem stated in the next section. To ease the presentation, assume that only the state constraints (3.4) can be relaxed when there is no feasible solution to the MPC problem. Relaxable input constraints can be easily included. In addition, variable right hand sides of the inequality constraints (3.3) and (3.4) can also be easily be included in the same framework. However, if this is done, the above cited stability result is no longer valid.

When relaxing the constraints, it is assumed that the constraints have different priorities. With priority, we mean that minimizing the relaxation of a constraint with priority level i is infinitely more important than minimizing the relaxation of a constraint with priority level $i+1$. Let H_i and h_i denote the ith row of H and the ith element of h_j respectively, and assume that $\forall\, i, j, t$, $H_i x_{j|t} \leq h_i$ has higher priority than $H_{i+1} x_{j|t} \leq h_{i+1}$, and that $H_i x_{j+1|t} \leq h_i$ has higher priority than $H_i x_{j|t} \leq h_i$. The latter is equivalent to saying that, if possible, all violations of $H_i x_{j|t} \leq h_i$ will occur during the first part of the prediction horizon. Inserting the predictor (3.2) into (3.4), all state inequality constraints can be collected into the inequality $\Theta U \leq \theta(x_t)$, where $\theta(x_t) := h + \Upsilon x_t$, and $\Theta, h, \Upsilon :=$

$$\begin{bmatrix} H_1 A^{j_2-1}B & H_1 A^{j_2-2}B & \cdots & H_1 A^{j_2-N}B \\ H_1 A^{j_2-2}B & H_1 A^{j_2-3}B & \cdots & H_1 A^{j_2-N-1}B \\ \vdots & \vdots & \ddots & \vdots \\ H_1 A^{N-1}B & H_1 A^{N-2}B & \cdots & H_1 B \\ H_1 A^{N-2}B & H_1 A^{N-3}B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_1 B & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_{n_h} A^{j_2-1}B & H_{n_h} A^{j_2-2}B & \cdots & H_{n_h} A^{j_2-N}B \\ H_{n_h} A^{j_2-2}B & H_{n_h} A^{j_2-3}B & \cdots & H_{n_h} A^{j_2-N-1}B \\ \vdots & \vdots & \ddots & \vdots \\ H_{n_h} A^{N-1}B & H_{n_h} A^{N-2}B & \cdots & H_{n_h} B \\ H_{n_h} A^{N-2}B & H_{n_h} A^{N-3}B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_{n_h} B & 0 & \cdots & 0 \end{bmatrix}, \begin{bmatrix} h_1 \\ h_1 \\ \vdots \\ h_1 \\ h_1 \\ \vdots \\ h_1 \\ \vdots \\ h_{n_h} \\ h_{n_h} \\ \vdots \\ h_{n_h} \\ h_{n_h} \\ \vdots \\ h_{n_h} \end{bmatrix}, - \begin{bmatrix} H_1 A^{j_2} \\ H_1 A^{j_2-1} \\ \vdots \\ H_1 A^{N} \\ H_1 A^{N-1} \\ \vdots \\ H_1 A \\ \vdots \\ H_{n_h} A^{j_2} \\ H_{n_h} A^{j_2-1} \\ \vdots \\ H_{n_h} A^{N} \\ H_{n_h} A^{N-1} \\ \vdots \\ H_{n_h} A \end{bmatrix}$$

Let Θ_i denote the ith row of Θ, and $\theta_i(x_t)$ the ith element of $\theta(x_t)$. Θ and $\theta(x_t)$ are constructed such that $\Theta_i U \leq \theta_i(x_t)$ has higher priority than $\Theta_{i+1} U \leq \theta_{i+1}(x_t)$. The constraints on the control inputs over the whole prediction horizon can be described as $\Delta U \leq \delta$, where Δ is block diagonal and the N blocks on the diagonal are Ds, and δ is a vector with composed of N ds stacked over each other. To summarize, the constraints in the MPC optimization problem are given by

$$\Gamma U = \gamma x_t, \;\; \Delta U \leq \delta, \;\text{and}\; \Theta U \leq \theta(x_t).$$

Under the assumption that the set of non-relaxable constraints is consistent (or, $\{U \,|\, \Gamma U = \gamma x_t \text{ and } \Delta U \leq \delta\} \neq \emptyset$), the algorithm presented in (Vada, Slupphaug and Foss, 1999) minimizes the constraint violations ϵ_k in the following way: $\epsilon^* := (\epsilon_1^*, \ldots, \epsilon_{n_h j_2}^*)$, where ϵ_k^* is the optimal constraint violation corresponding to the state constraint with priority level k, is computed according to the following algorithm:

Algorithm 3.2.1

Step 1: Solve the LP problem $\epsilon_1^* = \min_{U,\epsilon_1} \epsilon_1$ subject to $\epsilon_1 \geq \Theta_1 U - \theta_1(x_t)$, $\Gamma U = \gamma x_t$, $\Delta U \leq \delta, \epsilon_1 \geq 0$. If $n_h j_2 > 1$ set $k = 2$ and go to Step 2, else stop.

Step 2: Solve the LP problem $\epsilon_k^* = \min_{U,\epsilon_k} \epsilon_k$ subject to $\epsilon_k \geq \Theta_k U - \theta_k(x_t)$, $\Gamma U = \gamma x_t$, $\Delta U \leq \delta$, $\Theta_i U \leq \theta_i(x_t) + \epsilon_i^*$, $i \in \mathbb{I}_{k-1}^+$, $\epsilon_k \geq 0$. Go to Step 3.

Step 3: If $k < n_h j_2$, set $k \leftarrow k + 1$, and go to Step 2, else stop.

Now, the following QP problem associated with the MPC is guaranteed to be feasible and is subsequently solved:

$$U^* = \arg\min_U U^T F U + 2U^T G x_t$$

subject to:

$$\begin{aligned} \Gamma U &= \gamma x_t \\ \Delta U &\leq \delta \\ \Theta_1 U &\leq \theta_1(x_t) + \epsilon_1^* \\ &\vdots \\ \Theta_{n_h j_2} U &\leq \theta_{n_h j_2}(x_t) + \epsilon_{n_h j_2}^* \end{aligned}$$

where

$$F := \begin{bmatrix} B^T\bar{Q}B + R & B^T A^T \bar{Q} B & \cdots & B^T A^{T^{N-1}} \bar{Q} B \\ B^T \bar{Q} A B & B^T \bar{Q} B + R & \cdots & B^T A^{T^{N-2}} \bar{Q} B \\ \vdots & \vdots & \ddots & \vdots \\ B^T A^{T^{N-1}} \bar{Q} B & B^T A^{T^{N-2}} \bar{Q} B & \cdots & B^T \bar{Q} B + R \end{bmatrix}, \quad G := \begin{bmatrix} B^T \bar{Q} A \\ B^T \bar{Q} A^2 \\ \vdots \\ B^T \bar{Q} A^N \end{bmatrix},$$

and, for open loop stable predictors

$$\bar{Q} := \sum_{i=0}^{\infty} A^{T^i} C^T Q C A^i,$$

while, for open-loop unstable predictors

$$\bar{Q} := \tilde{V}_s^T \Sigma \tilde{V}, \; \Sigma := \sum_{i=0}^{\infty} J_s^{T^i} V_s^T C^T Q C V_s J_s^i$$

(Muske and Rawlings, 1993).

We now turn our attention to a more efficient way of determining minimum constraint violations than using Algorithm 3.2.1. Note that the result of

Algorithm 3.2.1 is straightforwardly associated with the solution of a so-called preemptive multi-objective linear program (Sherali, 1983). This fact will in Section 3.4 be exploited to design a single-objective LP problem such that ϵ^* is part of the solution to this LP, and such that the same LP can be used to compute ϵ^* for all x_t where there exists a solution to the non-relaxable constraints. This leads to significantly less on-line computational complexity than solving the sequence of $n_h j_2$ LP problems required by Algorithm 3.2.1.

3.2.2 Definition of the optimal weight design problem (OWDP)

Consider the following LP problem:

$$\xi^*(p) := (y^*(p), z^*(p)) := \underset{y,z}{\operatorname{argmin}}\ \tilde{c}^T z \tag{3.6a}$$

subject to:

$$\begin{array}{ll} G^1 y = g^1(p), & g^1(p) := g^{10} + g^{11}p \\ G^2 y \leq g^2(p), & g^2(p) := g^{20} + g^{21}p \\ G^3 y \leq g^3(p) + z, & g^3(p) := g^{30} + g^{31}p \\ y \ \geq 0 & \\ z \ \geq 0 & \end{array} \tag{3.6b}$$

where $y \in \mathbb{R}^{n_y}$, $z \in \mathbb{R}^{m_3}$, $\tilde{c} \in \mathbb{R}^{m_3}$, $G^1 \in \mathbb{R}^{m_1 \times n_y}$, $G^2 \in \mathbb{R}^{m_2 \times n_y}$, $G^3 \in \mathbb{R}^{m_3 \times n_y}$, $g^{i0} \in \mathbb{R}^{m_i}$, $g^{i1} \in \mathbb{R}^{m_i \times n_p}$, $i \in \mathbb{I}_3^+$, and $p \in \mathcal{P} \subset \mathbb{R}^{n_p}$, where $\mathcal{P}$ is the set of all p such that there exists an optimal solution to (3.6) , i.e.

$$\mathcal{P} = \{p \in \mathbb{R}^{n_p} \mid Y^{1,2}(p) \neq \emptyset\}, \tag{3.7}$$

where

$$Y^{1,2}(p) := \{y \geq 0 \mid G^1 y = g^1(p) \text{ and } G^2 y \leq g^2(p)\}. \tag{3.8}$$

Note that it is not assumed that $\mathcal{P}$ is bounded. Let $Z(p)$ denote the set of non-negative zs such that the set of ys satisfying (3.6b) is nonempty, i.e.

$$Z(p) = \left\{z \geq 0 \mid Y^{1,2}(p) \cap Y^3(p, z) \neq \emptyset\right\}, \tag{3.9}$$

where

$$Y^3(p, z) := \{y \geq 0 \mid G^3 y \leq g^3(p) + z\}. \tag{3.10}$$

The following assumptions are made throughout the rest of this paper:

A1: $\mathcal{P}$ is nonempty.

A2: G^1 has rank m_1.

Note that under A1, $\forall p \in \mathcal{P}$, $Z(p) \neq \emptyset$ (since $\forall p \in \mathcal{P}$, $Y^{1,2}(p) \neq \emptyset$ and $\forall p \in \mathcal{P}$, $\exists z \geq 0$ such that $Y^3(p,z) \neq \emptyset$). The necessity of A2 will become clear in Section 3.3.

Before presenting the problem definition, consider the following definition. The relation between this definition and Algorithm 3.2.1 becomes clear in Section 3.2.3.

DEFINITION 3.1
Given $X \subset \mathbb{R}^n$, $x^* \in X$ is its lexicographic minimum if $x_i^* < x_i$ for all $x \in X$ such that $x \neq x^*$, where $i \in \mathbb{I}_n^+$ is the index to the first element where x and x^* are different.

Let $z^o(p)$ denote the lexicographic minimum of $Z(p)$. Since $\forall p \in \mathcal{P}$, $Z(p) \neq \emptyset$ and closed, and $z \geq 0$, $z^o(p)$ exists $\forall p \in \mathcal{P}$. The following problem can now be stated:

Optimal weight design problem (OWDP) *Design the weight vector $\tilde{c}$ in (3.6a) such that $\forall p \in \mathcal{P}$, $z^*(p) = z^o(p)$, i.e. such that the z-part of any optimal solution to the LP (3.6) is equal to the lexicographic minimum of $Z(p)$.*

At this point, it is assumed that $\tilde{c}$ exists. This will be proved later on. If, for some reason, one only wants to consider ps which belong to a set $\mathcal{M} \subset \mathcal{P}$, this can be incorporated in the OWDP by replacing $\mathcal{P}$ with $\mathcal{P} \cap \mathcal{M}$. Due to the way we solve the OWDP in Section 3.4, it is necessary to assume that $\mathcal{M}$ is polyhedral.

The LP problem defined by (3.6) can be rewritten into standard form:

$$\left.\begin{aligned} &\min_x c^T x \\ &\text{subject to:} \quad \begin{cases} Ax &= b(p) \\ x &\geq 0 \end{cases} \\ A &:= \begin{bmatrix} G^1 & 0_{m_1 \times m_2} & 0_{m_1 \times m_3} & 0_{m_1 \times m_3} \\ G^2 & I_{m_2} & 0_{m_2 \times m_3} & 0_{m_2 \times m_3} \\ G^3 & 0_{m_3 \times m_2} & I_{m_3} & -I_{m_3} \end{bmatrix}, \\ b(p) &:= \Big(g^1(p),\, g^2(p),\, g^3(p) \Big), \\ c &:= \Big(0_{n_y + m_2 + m_3},\, \tilde{c} \Big), \\ x &:= \Big(y,\, v,\, w,\, z \Big), \end{aligned}\right\} \tag{3.11}$$

where $v \in \mathbb{R}^{m_2}$ and $w \in \mathbb{R}^{m_3}$. Due to Assumption A2, A has full row rank. From now on, A, B and D are not related to their definitions in Section 3.2.1.

3.2.3 Relation between the OWDP and the prioritized MPC problem

By inspection of Algorithm 3.2.1, we observe that ϵ^* is the lexicographic minimum of all possible $\epsilon = (\epsilon_1, \cdots, \epsilon_{n_h j_2})$ making the set of U which satisfy the following set of equalities/inequalities

$$\begin{aligned} \Gamma U &= \gamma x_t \\ \Delta U &\leq \delta \\ \Theta U &\leq h + \Upsilon x_t + \epsilon \\ \epsilon &\geq 0 \end{aligned} \tag{3.12}$$

nonempty. Note that the only variables in (3.12) are U and ϵ.

The constraints in (3.6b) includes the inequality $y \geq 0$, but in (3.12), a corresponding inequality, $U \geq 0$, is not explicitly present. However, (3.12) can easily be transformed to include such an inequality as follows: For simplicity, assume that U is bounded from below[2], i.e.

$$U \geq U^{\min}, \tag{3.13}$$

which, by the way, always will be the case in an MPC problem, since each element of U is related to a physical quantity. Next, by defining $\overline{U} := U - U^{\min}$, the constraints in (3.12) and (3.13) can be transformed to the following set of constraints:

$$\begin{aligned} \Gamma\overline{U} &= \overline{\gamma} + \gamma x_t & \overline{\gamma} &:= -\Gamma U^{\min} \\ \Delta\overline{U} &\leq \bar{\delta} + \delta & \bar{\delta} &:= -\Delta U^{\min} \\ \Theta\overline{U} &\leq \bar{h} + h + \Upsilon x_t + \epsilon & \bar{h} &:= -\Theta U^{\min} \\ \epsilon &\geq 0 \\ \overline{U} &\geq 0 \end{aligned} \tag{3.14}$$

Now, let $p = x_t$, $z = \epsilon$, $y = \bar{U}$, $G^1 = \Gamma$, $G^2 = \Delta$, $G^3 = \Theta$, $g^{10} = \overline{\gamma}$, $g^{11} = \gamma$, $g^{20} = \bar{\delta} + \delta$, $g^{21} = 0_{n_d N \times n}$, $g^{30} = \bar{h} + h$ and $g^{31} = \Upsilon$ in (3.6b), then (3.6b) and (3.12) are equivalent.

Note that since $\delta > 0$, assumption A1 holds if the pair (A, B) is stabilizable and $N \geq \max(n_u, 1)$, where n_u is the number of unstable modi of (3.2) . Further, note that A1 implies that A2 holds due to the following: If Γ does not have full row rank, then there are repeated equalities among the equalities $\Gamma\overline{U} = \overline{\gamma} + \gamma x_t$. Thus, full row rank of Γ (i.e. assumption A2) can be obtained by removing such repeated equalities.

Finding a solution to the OWDP will, at each sample, reduce the sequence of $n_h j_2$ LP problems in Algorithm 3.2.1 into a single *off-line* designed LP problem. This fact is the motivation behind stating the OWDP.

[2]This assumption does not reduce the degree of generality, since an unbounded variable U_i can in be substituted by $U_i = V_i - W_i$, $V_i \geq 0$, $W_i \geq 0$.

3.3 Existence of a solution to the OWDP

In this section, assume that the weights $\tilde{c}_i$ are defined via their consecutive ratios r_i as follows: given $\tilde{c}_{m_3} > 0$, $\tilde{c}_i := r_i\tilde{c}_{i+1}$, $r_i > 0$, $i \in \mathbb{I}^+_{m_3-1}$. The rationale behind this definition is: Assume that corresponding to a fixed $p \in \mathcal{P}$, a set of ratios $r^*(p)$ exists such that for *all* $r_i \geq r_i^*(p)$, the associated weights solve the OWDP for this particular p. Next, assume that such $r_i^*(p)$s exist for all $p \in \mathcal{P}$, and let $\hat{r}_i^*$ denote the maximum of $r_i^*(p)$ over $\mathcal{P}$ (existence of this maximum is proved below). Let

$$\tilde{C}(\bar{r}) := \{\tilde{c} \,|\, \tilde{c}_i = r_i\tilde{c}_{i+1}, \tilde{c}_{m_3} > 0, r_i > \bar{r}_i, i \in \mathbb{I}^+_{m_3-1}\}. \tag{3.15}$$

Thus, any $\tilde{c} \in \tilde{C}(\hat{r}^*)$ solves the OWDP.

At first glance, one might believe that it is sufficient only to find ratios $r_i(p)$, $i \in \mathbb{I}^+_{m_3-1}$, such that the corresponding weights solve the OWDP for this particular p, and then maximize *these* ratios over $\mathcal{P}$. However, this is not generally sufficient, since the ratios have to be sufficiently large to guarantee that such a maximization will work. Example 3.1 in Section 3.C illustrates this. In this section it is shown that such ratios exist, and in Section 3.4 we present a solution strategy to the OWDP.

Theorem 3.1 below states the existence of a solution to the OWDP. It is, in the present setting, an extension to the main result in (Sherali, 1983), since it considers *varying* right hand sides of the constraints. This is of vital importance in the context of *on-line* prioritized infeasibility handling in linear MPC.

THEOREM 3.1
Assume A1 and A2. Then there exist ratios $\hat{r}^* > 0$ such that $\forall \tilde{c} \in \tilde{C}(\hat{r}^*)$, the following is true: $\forall p \in \mathcal{P}$, $x^*(p) := (y^*(p), v^*(p), w^*(p), z^*(p))$ is an optimal solution to (3.11) if and only if $z^*(p) = z^o(p)$.

Before presenting the proof of Theorem 3.1, some useful results and definitions which are used in the proof (directly or indirectly) are stated.

THEOREM 3.2
Assume A1 and A2, and assume that $p \in \mathcal{P}$ is fixed such that $b = b(p)$ is fixed. Then there exist ratios $r^*(p) > 0$ such that $\forall \tilde{c} \in \tilde{C}(r^*(p))$, $x^*(p)$ is optimal to (3.11) if and only if $z^*(p) = z^o(p)$.

Proof: The proof follows directly from (Sherali, 1983, Theorem 2.1). □

DEFINITION 3.2
A square matrix B consisting of m linearly independent columns of $A \in \mathbb{R}^{m \times n}$ is called a *basis* for $\mathbb{R}^m$. If all $n - m$ components of x not associated with columns of B are set equal to zero, the solution to $Ax = b$ is called a *basic solution*. If, in addition, $x \geq 0$, it is called a *basic feasible solution* and a corresponding basis is called a *feasible basis*. The components of x associated with columns of B are called *basic variables*.

A fundamental prerequisite for the way we follow to solve the OWDP is existence of a basic solution $x(p) = (y(p), v(p), w(p), z(p))$ such that $z(p)$ is equal to the lexicographic minimum of $Z(p)$. This is the content of Lemma 3.1.

LEMMA 3.1
Assume A1, A2, and that $p \in \mathcal{P}$ is fixed such that $b = b(p)$ is fixed. Then x^o, which denotes the lexicographic minimum of the set $X := \{x \,|\, x \geq 0, Ax = b\}$, is a basic solution to $Ax = b$. Furthermore, let $Z'(p) := \{z \in Z(p) \mid \exists y, v, w \text{ such that } x = (y, v, w, z) \geq 0 \text{ is a basic solution to } Ax = b\}$. Then the lexicographic minimum of $Z'(p)$ is equal to $z^o(p)$.

Proof: The first part of the lemma follows from (Yu and Zeleny, 1975, Theorem 2.3). Note that A1 and A2 imply that x^o in Lemma 3.1 exists. The second part of the lemma follows from the first part by reordering the columns of A (and elements in x) such that the columns of A associated with z become the first columns of A (elements in x). □

Let $\mathcal{P}_B$ denote the set of all $p \in \mathcal{P}$ where B is a feasible basis to (3.11), i.e.

$$\mathcal{P}_B = \{p \in \mathcal{P} \mid B^{-1}b_0 + B^{-1}b_1 p \geq 0\}, \tag{3.16}$$

where $b_0 := (g^{10}, g^{20}, g^{30})$, $b_1 := (g^{11}, g^{21}, g^{31})$. The following lemma states that if, for a $p \in \mathcal{P}_B$, the corresponding basic solution $x(p)$ has the property that $z(p) = z^o(p)$, then this property holds for *all* $p \in \mathcal{P}_B$.

LEMMA 3.2
Assume A1 and A2, let B be a basis to (3.11), and let $x(p) = (y(p), v(p), w(p), z(p))$ be the corresponding basic solution. Then the following implication holds:

$$(\exists p \in \mathcal{P}_B,\ z(p) = z^o(p)) \Rightarrow (\forall p \in \mathcal{P}_B,\ z(p) = z^o(p)) \tag{3.17}$$

Proof: Follows from (Gal, 1995, p. 179). □

Finally, we state the proof of Theorem 3.1:

Proof of Theorem 3.1:

Consider the following candidates for the ratios $\hat{r}_i^*$ in the theorem

$$\hat{r}_i^* := \sup_{p \in \mathcal{P}} r_i^*(p), \quad i \in \mathbb{I}_{m_3-1}^+, \tag{3.18}$$

where the $r_i^*(p)$s are associated with Theorem 3.2. First we show that $\hat{r}_i^*$, $i \in \mathbb{I}_{m_3-1}^+$ are finite. This is indeed the case since we only have to consider a finite number of ps in order to compute the supremum in (3.18). This follows from the facts that *i*) there is only a finite number of bases, *ii*) for a given $p \in \mathcal{P}$, there will always be a basis B such that the corresponding basic solution solves the problem (Lemma 3.1), and *iii*) the same $r^*(p)$ (finite) solves the OWDP for all $p \in \mathcal{P}_B$ (Lemma 3.2).

Now, let $p \in \mathcal{P}$ be arbitrary, and let $\tilde{c} \in \tilde{\mathcal{C}}(\hat{r}^*)$, where $\hat{r}^*$ is computed as described above. Since $\tilde{c} \in \tilde{\mathcal{C}}(\hat{r}^*)$ implies that $\tilde{c} \in \tilde{\mathcal{C}}(r^*(p))$, where $r^*(p)$ is associated with Theorem 3.2, it follows from Theorem 3.2 that if this $\tilde{c}$ is used in (3.11), $z^*(p) = z^o(p)$. □

3.4 A solution to the OWDP

In the previous section existence of a solution to the OWDP was proved. In the following we will show how it can be solved. In the OWDP it is sufficient only to consider ps in the set $\mathcal{P} \setminus \mathcal{P}^0$, where $\mathcal{P}^0 := \{p \in \mathcal{P} \,|\, (3.6) \text{ has a solution with } z^*(p) = 0\}$, since $\forall p \in \mathcal{P}^0$, clearly any $\tilde{c} > 0$ renders $z(p)^* = z^o(p) = 0$. The basic idea behind the solution approach is as follows: Given a set of bases $\mathcal{B}$ such that $\mathcal{P} \setminus \mathcal{P}^0 \subseteq \cup_{B \in \mathcal{B}} \mathcal{P}_B$, where each basis $B \in \mathcal{B}$ has the following property: $\forall p \in \mathcal{P}_B$, $x(p) = (y(p), v(p), w(p), z(p))$, the basic solution to $Ax = b(p)$ corresponding to B, satisfies $z(p) = z^o(p)$. The weight vector $\tilde{c}$ in (3.11) is then designed such that the basic solutions corresponding to each of the bases $B \in \mathcal{B}$ are optimal when they are feasible. This ensures that $\forall p \in \mathcal{P}$, there always exists an optimal basic solution $x^*(p)$ to (3.11) such that $z^*(p) = z^o(p)$. Note that this does not imply that the OWDP is solved, since for a given p there might exist *alternative optimal solutions* to (3.11) that does *not* satisfy $z^*(p) = z^o(p)$. Hence, in order to solve the OWDP, $\tilde{c}$ must be designed such that $\forall p \in \mathcal{P}$, *each* optimal solution $x^*(p)$ to (3.11) satisfies $z^*(p) = z^o(p)$. In Section 3.4.1, this solution approach to the OWDP is described in detail, and in Section 3.4.2 we present an algorithm which can be used to compute $\mathcal{B}$.

3.4.1 Computing the weights

In this section, assume that the set of bases $\mathcal{B}$ (defined above) is given. Before presenting the solution approach, we need some definitions:

DEFINITION 3.3
A vector $x \neq 0$ is said to be *lexicographically positive* if the first nonzero component of x is positive. x is said to be *lexicographically nonnegative* if it is lexicographically positive or $x = 0$. Similarly, $x \neq 0$ is said to be *lexicographically negative* if the first nonzero component of x is negative. x is said to be *lexicographically nonpositive* if it is lexicographically negative or $x = 0$. $x \in \mathbb{R}^n$ is said to be *lexicographically less* than $x' \in \mathbb{R}^n$ if $x - x'$ is lexicographically negative.

Consider the following parametric preemptive multi-objective linear program (parametric PMOLP) defined by

$$\begin{aligned} &\min_x \; Wx \\ &\text{subject to } x \in X(p) := \{x \in \mathbb{R}^{n_y+m_2+2m_3} \mid Ax = b(p),\; x \geq 0\} \end{aligned} \tag{3.19}$$

where $W := [0_{m_3\times(n_y+m_2+m_3)}, I_{m_3}]$, and where there exists a lexicographic ordering among the objectives (i.e. elements of Wx), that is, $\sum_{j=1}^{n_y+m_2+2m_3} W_{i,j}x_j$ has higher priority than $\sum_{j=1}^{n_y+m_2+2m_3} W_{i+1,j}x_j$. Note that the adjective "parametric" is associated with the parameter p.

DEFINITION 3.4
For a fixed $p \in \mathcal{P}$, an optimal solution to the associated PMOLP is called a *preemptive optimal solution.* If the basic solution corresponding to a basis B is a preemptive optimal solution, B is called a *preemptive optimal basis.*

Note that if $x^*(p) = (y^*(p), v^*(p), w^*(p), z^*(p))$ is a preemptive optimal solution to (3.19), then there does not exist any $x \in X(p)$ such that Wx is lexicographically less than $Wx^*(p)$. Since $Wx = z$, due to the definition of W, and due to the lexicographic ordering among the objectives in (3.19), it follows that the z-part of a preemptive optimal solution to the parametric PMOLP is equal to $z^o(p)$. Thus, by using the relations described in Section 3.2.3, for any $p \in \mathcal{P}$, the optimal cost of (and the z-part of any optimal solution to) (3.19) is equal to the result of Algorithm 3.2.1.

Now, for a fixed $p \in \mathcal{P}$, let B be a preemptive optimal basis to the associated PMOLP, and let $x(p) = (y(p), v(p), w(p), z(p))$ be the corresponding basic solution, i.e. $z(p) = z^o(p)$. Further, let $x_B(p) \in \mathbb{R}^{m_1+m_2+m_3}$ and $x_D(p) \in \mathbb{R}^{n_y-m_1+m_3}$ denote the basic and nonbasic variables of $x(p)$, respectively. Moreover, let $c_B \in \mathbb{R}^{m_1+m_2+m_3}$ and $c_D \in \mathbb{R}^{n_y-m_1+m_3}$ denote the elements of c (defined in (3.11)) corresponding to $x_B(p)$ and $x_D(p)$, respectively. Based on the preemptive optimal basis B, sufficient conditions on the weights $\tilde{c}_i > 0$ to ensure nonnegativity of the relative cost vector $r_D^T := c_D^T - c_B^T B^{-1} D$ will now be derived. (Nonnegativity of r_D in addition to feasibility of $x(p)$ is equivalent to optimality of $x(p)$ (Luenberger, 1989).) One may write the expression for r_D as

$$r_D = \begin{bmatrix} I_{n_y - m_1 + m_3} & -D^T B^{-T} \end{bmatrix} \begin{bmatrix} c_D \\ c_B \end{bmatrix}. \tag{3.20}$$

Since $c_i = 0$, $i \in \mathbb{I}^+_{n_y+m_2+m_3}$, the corresponding elements in c_D and c_B in (3.20) can be removed along with the corresponding columns of $[I_{n_y-m_1+m_3}, -D^T B^{-T}]$. By exchanging the columns of the resulting matrix, the nonnegativity condition on r_D is equivalent to

$$r_D = S_B \tilde{c} \geq 0, \ S_B \in \mathbb{R}^{(n_y - m_1 + m_3) \times m_3}. \tag{3.21}$$

where the ith column of S_B is equal to the column of $[I_{n_y-m_1+m_3}, -D^T B^{-T}]$ corresponding to the element of (c_D, c_B) where $c_{n_y+m_2+m_3+i}$ $(= \tilde{c}_i)$ is located. Note that the subscript B indicates that S_B is computed with respect to the basis B, and that S_B is independent of p. If $\tilde{c}$ satisfies (3.21), the solution corresponding to the preemptive optimal basis B is also an optimal solution to (3.11), since, for the given $p \in \mathcal{P}$, B is a feasible basis to (3.19). In order to solve the OWDP, $\tilde{c}$ must be chosen such for all $p \in \mathcal{P}$, each alternative solution to (3.11) is also a preemptive optimal solution to the PMOLP. Lemma 3.4 below shows how such a $\tilde{c}$ can be computed, but first we present Lemma 3.3 and Definition 3.5, which is used in the proof of Lemma 3.4:

LEMMA 3.3
Assume A1 and A2 and let $p \in \mathcal{P}$ be fixed such that $b = b(p)$ is fixed. Further, let B be a preemptive optimal basis to the associated PMOLP. Then each row of the corresponding S_B is lexicographically nonnegative.

Proof: Due to A1, A2 and Lemma 3.1, a B exists $\forall p \in \mathcal{P}$. Let the ith row of W be denoted $\hat{e}_i$. In (Sherali, 1983, Theorem 2.1.) it is shown (for arbitrary W) that if there exists a solution to the associated PMOLP, then there exists a real scalar $M_0 > 0$ such that for any $M \geq M_0$, $x^*(p)$ is an optimal solution to

$$\begin{array}{ll} \min \ \tilde{c}^T x, & \tilde{c} = \sum_{i=1}^{q} M^{q-i} \hat{e}_i \\ \text{subject to } x \in X := \{x \mid Ax = b, \ x \geq 0\} \end{array} \tag{3.22}$$

if and only if it is an optimal solution to the associated PMOLP. Note that $\hat{e}_i = e_{n_y+m_2+m_3+i}$, $i \in \mathbb{I}^+_{m_3}$. Thus, for the particular p, if, in (3.11),

$c = \sum_{i=1}^{m_3} M^{m_3-i} e_{n_y+m_2+m_3+i}$, $M \geq M_0$ (i.e. $\tilde{c} = \sum_{i=1}^{m_3} M^{m_3-i} e_i = (M^{m_3}, M^{m_3-1}, \cdots 1)$), then these weights render $z^*(p) = z^o(p)$. Thus, let $\tilde{c} = (M^{m_3}, M^{m_3-1}, \cdots 1)$.

Assume that Lemma 3.3 is false, i.e. assume that there exist $i, j \in \mathbb{I}_{m_3}^+$ such that $(S_B)_{i,j} < 0$ is the first nonzero element in row i of S_B. Let $x'(p)$ be the basic solution to (3.11) corresponding to B. Then it follows that there exists an $M' > 0$ such that if $M \geq M'$, then $S_B \tilde{c} \ngeq 0$, i.e., $x'(p)$ will not be optimal to (3.11) if the ratio between consecutive weights is large enough. This is due to the following: We have that $(S_B \tilde{c})_i = \sum_{k=j}^{m_3} (S_B)_{i,k} M^{m_3-k}$. As M increases, $(S_B)_{i,j} M^{m_3-j}$ dominates this sum, and for some sufficiently large M, say M', $(S_B \tilde{c})_i$ becomes negative for all $M \geq M'$, which contradicts the result from (Sherali, 1983) stated above. □

Definition 3.5

(Steuer, 1986, p. 216) Let the *relative cost matrix* corresponding to the basis B be defined as

$$R_D := W_D - W_B B^{-1} D, \tag{3.23}$$

where D consists of the columns of A not in B and W_B (W_D) consists of the columns of W corresponding to the basic (nonbasic) variables.

Note that the ith row of R_D is the relative cost vector corresponding to the cost function $\sum_{j=1}^{n_y+m_2+2m_3} W_{i,j} x_j$.

Lemma 3.4

Assume A1 and A2, and assume given a set of preemptive optimal bases $\mathcal{B}$ such that $\mathcal{P} \setminus \mathcal{P}^0 \subseteq \cup_{B \in \mathcal{B}} \mathcal{P}_B$. Then there exists an optimal solution $(\tilde{c}^*, \bar{c}^*)$ to the following LP problem:

$$\min_{\tilde{c}, \bar{c}} \bar{c}$$

subject to:

$$\begin{aligned} \sum_{j=1}^{m_3} (S_B)_{i,j} \tilde{c}_j &\geq \underline{r}_D, \; i \in \mathbb{I}_{n_y-m_1+m_3}^+ \setminus \mathcal{Z}_B, \; B \in \mathcal{B} \\ \tilde{c}_i &\geq \underline{c}, \quad i \in \mathbb{I}_{m_3-1}^+ \\ \tilde{c}_i &\leq \bar{c} \quad i \in \mathbb{I}_{m_3-1}^+ \end{aligned} \tag{3.24}$$

where $\underline{c}, \underline{r}_D \in \mathbb{R}^+$ and $\mathcal{Z}_B := \{i \in \mathbb{I}_{n_y-m_1+m_3}^+ \mid \forall j \in \mathbb{I}_{m_3}^+, (S_B)_{i,j} \geq 0\}$, i.e. $\mathcal{Z}_B$ is the index set to all rows of S_B containing nonnegative elements only. Moreover, $\tilde{c}^*$ solves the OWDP, that is: if $\tilde{c} = \tilde{c}^*$ in (3.11), $\forall p \in \mathcal{P}$, $x(p) = (y(p), v(p), w(p), z(p))$ is an optimal solution to (3.11) if and only if $z(p) = z^o(p)$ *(i.e. if and only if $x(p)$ is a preemptive optimal solution to (3.19)).*

Proof: See Section 3.A.1. □

Since numerical problems when solving (3.11) may occur if the ratio between the largest and smallest element of $\tilde{c}$ is large, the cost function in (3.24) is designed to minimize this ratio. The constants $\underline{c}$ and $\underline{r}_D$ can also be tuned with the same objective in mind.

3.4.2 Finding a set of preemptive optimal bases

In this section we propose an algorithm (Algorithm 3.4.2) which computes a finite set of bases $\mathcal{B}$ such that $\mathcal{P} \subseteq \cup_{B \in \mathcal{B}} \mathcal{P}_B$, where each basis $B \in \mathcal{B}$ has the following property: $\forall p \in \mathcal{P}_B$, $x(p) = (y(p), v(p), w(p), z(p))$, the basic solution to $Ax = b(p)$ corresponding to the basis B, satisfies $z(p) = z^o(p)$, that is, for each basis $B \in \mathcal{B}$, $\forall p \in \mathcal{P}_B$, B is a preemptive optimal basis to (3.19). Note that Algorithm 3.4.2 computes a set of preemptive optimal bases $\mathcal{B}$ which covers $\mathcal{P}$, while in Section 3.4.1, it is required that $\mathcal{B}$ covers $\mathcal{P} \setminus \mathcal{P}^0$. However, this does not influence the resulting weights $\tilde{c}$ computed by solving (3.24), since $\forall p \in \mathcal{P}^0$ any $\tilde{c} > 0$ solves the OWDP.

Algorithm 3.4.2 is based on an algorithm stated in (Gal, 1995, Chapter IV-3-1), which solves the following problem: Given a general (single-objective) parametric LP problem such as (3.11). Find a set of polytopes $\mathcal{P}_B$ that cover $\mathcal{P}$ such that $\forall p \in \mathcal{P}_B$, the corresponding basis B is optimal to (3.11). The basic idea is to use a dual simplex step to pass from one region $\mathcal{P}_{B'}$ to a neighboring region $\mathcal{P}_{B''}$, where $\forall p \in \mathcal{P}_{B'}$ ($\mathcal{P}_{B''}$) B' (B'') is an optimal basis to (3.11). This procedure is continued until all optimal bases are computed.

Note that in the problem considered in (Gal, 1995, Chapter 4), there is a scalar cost function. Since we in the present problem considers a vectorial cost function with a lexicographic ordering among the elements, a modification of this algorithm is required. Before presenting Algorithm 3.4.2, we need some results and definitions. The following lemma provides a characterization of the relative cost matrix corresponding to a preemptive optimal basis. A similar result is also stated in (Korhonen and Halme, 1996) (without a proof).

LEMMA 3.5
Fix $p \in \mathcal{P}$ and let B be a basis to (3.19). B is a preemptive optimal basis to (3.19) if and only if B is feasible and each column of the corresponding relative cost matrix R_D is lexicographically nonnegative.

Proof: See Section 3.A.2. □

Theorem 3.3 below presents the dual simplex step in the sense of PMOLP, which can be used to pass from one preemptive optimal basis to other preemptive optimal bases. The main point is to ensure that each column of the relative cost matrix corresponding to each of these bases are lexicographically nonnegative.

THEOREM 3.3
Given a $p \in \mathcal{P}$ and a corresponding preemptive optimal basis B to (3.19) (thus $p \in \mathcal{P}_B$). Given a $p^0 \in \mathcal{P}_B$ and an $r \in \mathbb{I}^+_{m_1+m_2+m_3}$ such that $(x_B(p^0))_r = 0$ and $\forall i \in \mathbb{I}^+_{m_1+m_2+m_3} \setminus \{r\}$, $(x_B(p^0))_r > 0$, i.e. p^0 is on one of the facets of $\mathcal{P}_B$ (recall that $x_B(p^0) = B^{-1}b(p^0)$). Let $p^\perp$ be a normal vector to this facet pointing out of $\mathcal{P}_B$. Then, if there exists a preemptive optimal basis to (3.19) for a $p = p' := p^0 + \varepsilon p^\perp$, with $\varepsilon \in \mathbb{R}^+$ infinitesimal small, the new basis computed by the following algorithm is a preemptive optimal basis to (3.19):

Algorithm 3.4.1 (Dual simplex step in the sense of PMOLP)
Note that to be consistent with the common notation in the LP literature, the symbols J, y, c, and z are redefined in this algorithm.

Step 1: Let $y_{i,j} := (B^{-1}A)_{i,j}$. If $\forall j \in \mathbb{I}^+_{n_y+m_2+2m_3}$, $y_{r,j} \geq 0$ there is no feasible solution to (3.19) when $p = p'$, stop. Else, set $k = 1$, $J_0^r := \{j \mid y_{r,j} < 0\}$ and go to Step 2.

Step 2: Let

$$J_k^r := \left\{ j \in J_{k-1}^r \mid \frac{z_{k,j} - c_{k,j}}{y_{r,j}} = \min_{i \in J_{k-1}^r} \frac{z_{k,i} - c_{k,i}}{y_{r,i}} \right\}.$$

where $z_{i,j} := c_B^i B^{-1} A_j$, $c_{i,j} := W_{i,j}$ and c_B^i consists of the elements of the ith row of W corresponding to the basic variables. Go to Step 3.

Step 3: If $|J_k^r| > 1$, and $k < m_3$ set $k \leftarrow k+1$, and go to Step 2. Else set q equal to one of the elements in J_k^r. Use y_{rq} as pivot element, i.e. form a new basis by replacing the rth column of B by the qth column of A. Stop.

Proof: See Section 3.A.3. Note that this theorem is a slight extension of a similar result stated in (Korhonen and Halme, 1996). However their proof is not complete since they do not prove feasibility of the new basis and that the optimality condition in Lemma 3.5 holds for *all* columns of R_D. □

Next, we define a measure for the distance between to bases (Gal, 1995, Chapter 4):

DEFINITION 3.6
Let ρ^k denote the set of indices to basic variables in the basic solution corresponding to B^k, where B^k is a basis to (3.11). ρ^i and ρ^j have the *distance* $\Delta = |\rho^i \setminus \rho^j|$.

The following two definitions are preemptive multi-objective extensions of similar definitions in (Gal, 1995):

DEFINITION 3.7
Given two bases B' and B'', $B' \neq B''$, to (3.19), and assume that there exist $p', p'' \in \mathcal{P}$ such that the basic solution corresponding to B' (B'') is a preemptive optimal solution to (3.19) if $p = p'$ ($p = p''$). B' and B'' are said to be *neighboring bases in the sense of PMOLP* if *i*) there exists a p^1 such that both B' and B'' are preemptive optimal bases to (3.19) if $p = p^1$, and *ii*) it is possible to pass from B' to B'' by a dual simplex step in the sense of PMOLP (see Algorithm 3.4.1), and vice versa.

Finally, we present Algorithm 3.4.2, which computes the set of preemptive optimal bases $\mathcal{B}$ such that $\mathcal{P} \setminus \mathcal{P}^0 \subseteq \cup_{B \in \mathcal{B}} \mathcal{P}_B$. Note that if one wants to consider an $\mathcal{M} \neq \mathbb{R}^{n_p}$ (see the comment after presenting the OWDP at the end of Section 3.2.2), this must be explicitly accounted for in this algorithm. However, if $\mathcal{M} = \mathbb{R}^{n_p}$, in the following, just replace $\mathcal{M}$ by $\mathbb{R}^{n_p}$. In Algorithm 3.4.2, let $\Phi(\rho^k)$ be the set of all index sets to basis variables such that $\rho^i \in \Phi(\rho^k)$ if and only if ρ^i is an index set to a neighboring basis in the sense of PMOLP to ρ^k satisfying $P_{B^i} \cap \mathcal{M} \neq \emptyset$. Further, at the kth iteration, let T_k be the set of all index sets to bases for which all corresponding neighboring basis in the sense of PMOLP have been computed, and let U_k be the set of all index sets to bases which are known to be neighboring bases in the sense of PMOLP to a basis in T_k, but whose corresponding set of all neighboring bases in the sense of PMOLP is not yet computed.

Algorithm 3.4.2
(Gal, 1995, Chapter IV-3)

Step 1: Find an arbitrary feasible solution $(x^*, p^*) \in \{(x, p) \mid Ax - b_1 p = b_0,$ $p \in \mathcal{M},\ x \geq 0\}$. If there is no feasible solution, $\mathcal{P} \cap \mathcal{M} = \emptyset$. Stop. Else, find a preemptive optimal basic solution to (3.19) with $p = p^*$. Such a solution can be computed using a sequential algorithm such as Algorithm 3.2.1. Let B^0 denote the corresponding basis. Set $T_0 = \{\rho^0\}$, and let $U_0 = \Phi(\rho^0)$. If $U_0 = \emptyset$, the algorithm is finished, stop. Else, set $k = 1$, and go to Step 2.

Step 2: Select a $\rho^k \in U_{k-1}$ with the smallest possible distance from ρ^{k-1}. In particular, if $\Phi(\rho^{k-1}) \cap U_{k-1} \neq \emptyset$, select a $\rho^k \in \Phi(\rho^{k-1}) \cap U_{k-1}$. Set $T_k = T_{k-1} \cup \{\rho^k\}$ and $U_k = U_{k-1} \cup \Phi(\rho^k) \setminus T^k$.

Step 3: If $U_k = \emptyset$, T_k contains a set of basis indices to preemptive optimal bases $\mathcal{B}$ such that $\mathcal{P} \cap \mathcal{M} \subseteq \cup_{B \in \mathcal{B}} \mathcal{P}_B$, stop. Else, set $k \leftarrow k+1$, and go to Step 2.

In Step 2 in Algorithm 3.4.2, one needs a procedure which computes the set of all neighboring bases in the sense of PMOLP to a given basis. In (Gal, 1995, Chapter 4), such a procedure is given for single-objective LPs. Algorithm 3.4.3 below is preemptive multi-objective extension of this algorithm:

Algorithm 3.4.3

Step 1: Let B be a preemptive optimal basis to (3.19) for a fixed $p \in \mathcal{P} \cap \mathcal{M}$, and let $\mathcal{I} \subseteq \mathbb{I}^+_{m_1+m_2+m_3}$ be the set of indices to rows in $B^{-1}A$ having at least one negative element.

Step 2: *For each $i \in \mathcal{I}$, compute*

$$s_i^* = \min_{p,s} \; s_i \tag{3.25a}$$

subject to

$$(p, s) \in \{(p, s) \mid s \geq 0, \; p \in \mathcal{M}, \; B^{-1}b_0 + B^{-1}b_1 p - s = 0\}. \tag{3.25b}$$

Let $\bar{\mathcal{I}}$ denote the set of all $i \subseteq \mathcal{I}$ such that $s_i^* = 0$. Note that each $i \in \bar{\mathcal{I}}$ corresponds to a facet of $\mathcal{P}_B$. (An efficient algorithm for solving (3.25), which exploits the nature of (3.25), is given in (Gal, 1995, Chapter II-2-2).)

Step 3: *For each $i \in \bar{\mathcal{I}}$, determine a possible pivot element according to the dual simplex step in the sense of PMOLP (Algorithm 3.4.1), and form the corresponding basis. This set of bases is the set of all neighboring bases in the sense of PMOLP to B.*

The proof that Algorithm 3.4.2 solves the problem of computing a finite set of bases $\mathcal{B}$ such that $\mathcal{P} \cap \mathcal{M} \subseteq \cup_{B \in \mathcal{B}} \mathcal{P}_B$, where each basis $B \in \mathcal{B}$ is a preemptive optimal basis to (3.19) is a trivial modification of the proof of the corresponding algorithm in (Gal, 1995).

As stated at the beginning of this section, Algorithm 3.4.2 computes a set of preemptive optimal bases $\mathcal{B}$ which covers $\mathcal{P}$ rather than $\mathcal{P} \setminus \mathcal{P}^0$, which is required to solve the OWDP with the solution strategy we propose in Section 3.4.1. (Recall that this difference does not influence the resulting weights.) If, in advance, it is known that the set $\mathcal{P} \setminus \mathcal{P}^0$ is connected, Algorithm 3.4.2 can be modified such that it does not consider bases B such that $\mathcal{P}_B$ and $\mathcal{P}^0$ are overlapping. By definition, two overlapping regions have at least one common interior point. This can be done in the following way: In Step 2 in Algorithm 3.4.2, let $\Phi(\rho^k)$ be the set of all index sets ρ^i satisfying the following two criteria: *i)* ρ^i is an index set to a neighboring basis in the sense of PMOLP to ρ^k satisfying $P_{B^i} \cap \mathcal{M} \neq \emptyset$, and *ii)* $\exists j \in \{n_y + m_1 + m_3 + 1, \dots, n_y + m_1 + 2m_3\}$ such that $j \in \rho^i$. Criteria *i)* is already present in Algorithm 3.4.2, while criteria *ii)* is added to rule out the basic solutions which do not have z_is as basic variables. The proof that this modification does not violate any of the assumptions in (Gal, 1995) is straightforward, thus, if $(\mathcal{P} \cap \mathcal{M})\setminus\mathcal{P}^0$ is connected, Algorithm 3.4.2 will continue until $(\mathcal{P} \cap \mathcal{M})\setminus\mathcal{P}^0$ is covered.

Note that $\mathcal{B}$ computed by Algorithm 3.4.2 is the set of *all* preemptive optimal bases, i.e. $B \in \mathcal{B}$ *if and only if* B is a preemptive optimal basis and $\mathcal{P}_B \cap \mathcal{M} \neq \emptyset$. Thus, some of the regions $\mathcal{P}_B$ will generally be overlapping, since, for a given $p \in \mathcal{P} \cap \mathcal{M}$, there might exist more than one preemptive optimal basis to (3.19). This is due to the fact that in Step 3 in Algorithm 3.4.1, if $|J^r_{m_3}| > 1$, there are more than one neighboring basis in the sense of PMOLP along the same facet. However, in order to solve the OWDP (according to Lemma 3.4) only a set of preemptive optimal bases such that $\mathcal{P} \cap \mathcal{M} \subseteq \cup_{B\in\mathcal{B}}\mathcal{P}_B$ is needed. In order to reduce the computational load when solving the OWDP, it suffices to find a subset $\tilde{\mathcal{B}}$ of $\mathcal{B}$ such that the corresponding polytopes $\mathcal{P}_B$ that cover $\mathcal{P}\cap\mathcal{M}$ are non-overlapping (note that there generally exists several such $\tilde{\mathcal{B}}$s). Existence of a $\tilde{\mathcal{B}}$ follows from Theorem 3.3, since, if there exists an optimal solution outside a facet of a given $\mathcal{P}_B$, a corresponding optimal basis B' (i.e. a neighboring basis in the sense of PMOLP) can be computed by Algorithm 3.4.1. Note that, in Step 3 in Algorithm 3.4.1, B' is obtained from B by pivoting on the y_{rq}th element, which is negative, and thus, $\mathcal{P}_B$ and $\mathcal{P}_{B'}$ are separated by a hyperplane. A $\tilde{\mathcal{B}}$ can be obtained from Algorithm 3.4.2 if the following modification is done: When computing the set of all neighbors in the sense of PMOLP to a given preemptive optimal basis (Algorithm 3.4.3), replace the cost matrix W in Algorithm 3.4.1 with $\hat{W} := [W^T, c^{aux}]^T$, and replace m_3 in Step 3 in Algorithm 3.4.1 by $m_3 + 1$, where the auxiliary cost vector

$c^{aux} \in \mathbb{R}^{n_y+m_2+2m_3}$ has the following property:

$$\forall B \in \mathcal{B}, r \in \mathbb{I}^{+}_{m_3+1}, i, j \in J_0^r, i \neq j, (r_D^{aux})_i / y_{r,i} \neq (r_D^{aux})_j / y_{r,j}, \tag{3.26}$$

where $(r_D^{aux})^T = (c_B^{aux})^T B^{-1} D - (c_D^{aux})^T$. The point is to ensure that the pivot element $y_{r,q}$ is uniquely determined in Step 3 in Algorithm 3.4.1. A c^{aux} satisfying (3.26) with probability 1 is obtained by setting each element of c^{aux} equal to a randomly selected positive real number. Given a preemptive optimal basis B and the corresponding $\mathcal{P}_B$, then this modification ensures that a neighbor along a certain facet of $\mathcal{P}_B$ is uniquely determined if there exists one. It follows from Theorem 3.3 that with this modification to Algorithm 3.4.3, $\mathcal{P} \cap \mathcal{M}$ will still be covered by the resulting set of bases.

For the single-objective case, Gal (1995) proposes the following method to compute a $\tilde{\mathcal{B}}$: when, in the single objective version of Algorithm 3.4.1, there are several possible neighboring bases along the same facet of a $\mathcal{P}_{B^k}$ corresponding to a given basis index set ρ^k, arbitrarily select one of them to enter U_k in Step 2 in Algorithm 3.4.2. However, experiments have shown that this strategy does not always work, and in Section 3.B an imaginary example is used to explain the reason for this. Moreover, according to Professor Tomas Gal (personal communication), his algorithm is indeed not worked out in all details for the cases when there are several optimal bases for a given $p \in \mathcal{P} \cap \mathcal{M}$. Since the single-objective case can be considered as a special case of the PMOLP, the approach presented above will solve the problem of overlapping polytopes in the single-objective case as well.

3.4.3 Algorithm to solve the OWDP

The following algorithm is a summary of the strategy we have proposed for solving the OWDP:

Algorithm 3.4.4
Given the constraints to the LP problem defined by (3.6).

Step 1: Define the corresponding A and $b(p)$ according to (3.11).

Step 2: Use Algorithm 3.4.2, with the above proposed modification, to compute a set of preemptive optimal bases $\tilde{\mathcal{B}}$ such that $\mathcal{P} \setminus \mathcal{P}^0 \subseteq \cup_{B \in \tilde{\mathcal{B}}} \mathcal{P}_B$ and such that $\forall B', B'' \in \tilde{\mathcal{B}}$, $B' \neq B''$, $\mathcal{P}_{B'}$ and $\mathcal{P}_{B''}$ are non-overlapping.

Step 3: One solution to the OWDP is given by a solution to the LP (3.24).

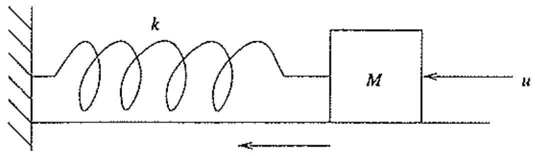

Figure 3.1: An idealized mass-spring system.

3.4.4 An alternative approach to solve the OWDP

As a first step towards developing an algorithm to solve the OWDP, we followed the same strategy of defining the weights $\tilde{c}_i$ via their consecutive ratios r_i as used when establishing the existence of a solution to the OWDP in Section 3.3. One such solution approach is presented in (Vada, Slupphaug and Johansen, 1999a). In Section 3.C it is shown how to compute ratios which will be less than or equal to the ratios obtained by using the strategy in (Vada, Slupphaug and Johansen, 1999a). Note that small ratios are desired in order to avoid numerical problems when solving the LP problem (3.6). However, the resulting $\tilde{c}$ obtained by using this solution approach generally gives larger ratios between the greatest and least element of $\tilde{c}_i$ than obtained by using Algorithm 3.4.4.

3.5 An example

In this section we solve the OWDP for an idealized mass-spring system which is illustrated in Figure 3.1. The spring is assumed to be linear, and the mass slides without any friction. There is a force u directed horizontally on the mass. It is assumed that both the position and the velocity of the mass are ideally measured, and that the spring constant $k = 1\ \frac{\text{N}}{\text{m}}$ and the mass $M = 1$ kg. By using exact discretization with sample time equal to 0.5 s, the system is given by the following equation:

$$x_{t+1} = A_d x_t + B_d u_t, \tag{3.27}$$

where

$$A_d = \begin{bmatrix} 0.8776 & 0.4794 \\ -0.4794 & 0.8776 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0.1224 \\ 0.4794 \end{bmatrix}.$$

$x_{t,1}$ and $x_{t,2}$ is the position and velocity of the mass at time t respectively. Let $N = j_2 = 5$. The input and state constraints are given by (3.3) and

(3.4) respectively, where

$$H = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \ h = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}, \ D = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \ d = \begin{bmatrix} 0.6 \\ 0.6 \end{bmatrix},$$

i.e. $n_h = 4$. The prioritization of the state constraints is as follows, in descending order:

i. Upper limit on the mass position, i.e. $x_{j|t,1} \leq 0.25, \ j > t$.

ii. Lower limit on the mass position, i.e. $x_{j|t,1} \geq -0.25, \ j > t$.

iii. Upper limit on the mass velocity, i.e. $x_{j|t,2} \leq 0.25, \ j > t$.

iv. Lower limit on the mass velocity, i.e. $x_{j|t,2} \geq -0.25, \ j > t$.

In addition to the above prioritization, $\forall i, j, t, \ H_i x_{j+1|t} \leq h_i$ has higher priority than $H_i x_{j|t} \leq h_i$. Thus, $m_3 = n_h j_2 = 20$, i.e. in the MPC optimization problem (described in Section 3.2.1) there are 20 constraints which can be relaxed in order to recover from infeasibility of (3.6), each having different priority levels. In the following, for notational simplicity, let $p = x_t$. Since both eigenvalues of A_d are equal to 1, $m_1 = 2$. Further, $m_2 = Nm = 5$, and $n_y = N = 5$. Thus, in (3.11), $\tilde{c} \in \mathbb{R}^{20\times 1}$, $A \in \mathbb{R}^{27\times 50}$, $b(p) \in \mathbb{R}^{27\times 1}$ and $x \in \mathbb{R}^{50\times 1}$. Figure 3.2 shows one way to cover $\mathcal{P} \setminus \mathcal{P}^0$ with a set of polytopes $\mathcal{P}_B$, where $\forall p \in \mathcal{P}_B, \ z(p) = z^o(p)$, where $x = (y(p), v(p), w(p), z(p)) \geq 0$ is the basic solution to $Ax = b(p)$ corresponding to the basis B. In the figure, there are 128 distinct $\mathcal{P}_B$s. These are computed by using Algorithm 3.4.2 in Section 3.4.2. The computation time was about 40 seconds on a Pentium 266MHz PC using MATLAB with NAG Foundation Toolbox. The area within the outer (i.e. largest) polytope shows the set of all initial states x_t where there exists a solution to (3.6), i.e. the set of all initial states $\mathcal{P}$ where the input constraints (3.3) in addition to the end point constraints (3.5) can be met by the proposed controller. The polytope enclosing the origin shows the set of all initial states x_t where there exists a solution to (3.6) with $z = 0$, i.e. the set of initial states $\mathcal{P}^0$ which does not require constraint violations. The area in-between those two polytopes, i.e. $\mathcal{P} \setminus \mathcal{P}^0$, are covered with polytopes having the following property: $\forall p \in \mathcal{P}_B, \ z(p) = z^o(p)$, where $p = x_t$.

The resulting weights computed by (3.24), with $\underline{c} = \underline{r}_D = 1.0$, are shown in Table 3.1. Note that the values of the resulting weights are non-intuitive in that one might expect that $\tilde{c}_1$ had to be significantly larger than $\tilde{c}_{20}$ in

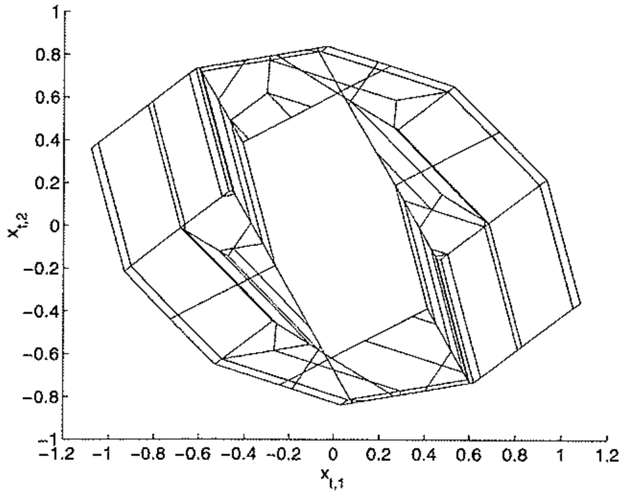

Figure 3.2: A partition of the state space.

i	1	2	3	4	5	6	7
$\tilde{c}_i^*$	1.000	1.000	1.000	78.160	78.160	1.000	1.000
i	8	9	10	11	12	13	14
$\tilde{c}_i^*$	1.000	78.160	78:160	1.000	5.747	2.086	2.086
i	15	16	17	18	19	20	
$\tilde{c}_i^*$	2.086	1.000	5.747	2.086	2.086	2.086	

Table 3.1: The weights solving the OWDP for the given example.

order to solve the OWDP, and in general that $\tilde{c}_i$ had to be much larger than $\tilde{c}_{i+1}$. Also note that the ratio between the largest and smallest elements of $\tilde{c}^*$ is only 78.16. To verify that $\tilde{c}$ in Table 3.1 indeed solves the OWDP, (3.11) was solved using these weights for 250000 different initial states x_t picked randomly from $\mathcal{P} \setminus \mathcal{P}^0$, and the solutions were compared with the corresponding solution computed by Algorithm 3.2.1. The solutions became equal for each initial state x_t.

3.6 Concluding remarks

Seeking an LP problem whose purpose is to minimize the violations of the relaxable constraints according to some prioritization in order to obtain a consistent set of constraints, we have shown that it is possible to design the weights of this LP such that the optimum becomes equal to the lexicographic minimum of the set of feasible constraint violations for all possible right hand sides of the constraints. This result is an extension of the main result in (Sherali, 1983) which considers the same mathematical problem when the right hand side of the constraints is fixed. Further, we present an algorithm to compute these weights. The first part of this algorithm computes a finite set of preemptive optimal bases such that all possible right hand sides of the constraints are associated with one of these bases. This part of the algorithm is an extension to a result in (Gal, 1995), which considers the same problem in the case when the weight vector to a scalar LP problem is given. The second part of the algorithm computes sufficient large weights corresponding to this set of bases.

Regarding MPC problems having different priority levels assigned to the constraints, this result can be used to compute the *optimal prioritized* constraint violations with only a small increase in on-line computational overhead, compared to the computational load of an MPC implementation without an infeasibility handler, and with a significant decrease in computational load compared to an infeasibility handler that sequentially removes single constraints to achieve feasibility.

3.A Some proofs

3.A.1 Proof of Lemma 3.4

In this proof, if K is a matrix, K_i denotes the ith column of K. First, existence of an optimal solution to (3.24) is established. From Theorem 3.1

it follows that there exists a feasible solution to (3.24) when $\underline{r}_D = 0$, and by using the fact that each row of S_B are lexicographically nonnegative (Lemma 3.3), it can be shown that there also exists a feasible solution for an arbitrary $\underline{r}_D > 0$. An optimal solution to (3.24) exists since the $\tilde{c}_i$s are bounded from below by non-strict inequalities.

Next, we prove the if and only if-part: Take a $p \in \mathcal{P}$, and let $B \in \mathcal{B}$ be a basis such that $p \in \mathcal{P}_B$. (Then, for all $p \in \mathcal{P}_B$, B is a preemptive optimal basis to (3.19)). Note that due to A1, A2 and Lemma 3.1, such a B exists for all $p \in \mathcal{P}$. Let $x^*(p) = (y^*(p), v^*(p), w^*(p), z^*(p))$ be the corresponding basic solution, i.e. $z^*(p) = z^o(p)$. Let $\mathcal{Z}_B^{\equiv 0} := \{i \in \mathbb{I}^+_{n_y - m_1 + m_3} \,|\, \forall j \in \mathbb{I}^+_{m_3}, (S_B)_{i,j} = 0\}$, i.e. $\mathcal{Z}_B^{\equiv 0} \subseteq \mathcal{Z}_B$ is the index set to all rows of S_B containing zeros only. First, assume that $\mathcal{Z}_B^{\equiv 0} = \emptyset$. Then by (3.24), the corresponding relative cost vector $S_B\tilde{c}^* > 0$, and hence $x^*(p)$ is a unique solution to (3.11). Since $x^*(p)$ is a preemptive optimal solution to (3.19), this proves the lemma when $\mathcal{Z}_B^{\equiv 0} = \emptyset$. Next, assume that $\mathcal{Z}_B^{\equiv 0} \neq \emptyset$, i.e. some of the elements of the relative cost vector $S_B\tilde{c}^*$ corresponding to B are equal to zero. Then $x^*(p)$ is not necessarily unique, and hence we have to prove that, when $\tilde{c} = \tilde{c}^*$, *all* alternative optimal solutions $x^*(p)$ to (3.11) are also preemptive optimal solutions to (3.19) and vice versa. The set of all optimal (basic as well as nonbasic) solutions to (3.11) consists of all feasible solutions to (3.11) in which the (nonbasic) variables $x_i^*(p)$ corresponding to non-zero relative cost coefficients, $(r_D)_i = (S_B\tilde{c}^*)_i$, are equal to zero (Murty, 1983, p. 139). Since $\underline{r}_D > 0$ in (3.24), then $(S_B\tilde{c}^*)_i = 0$ if and only if $i \in \mathcal{Z}_B^{\equiv 0}$ (i.e. the ith row of S_B contains zeros only). Let I_B and I_D denote the vectors of indices to the basic and nonbasic variables in $x^*(p)$ respectively. From the way S_B is composed (recall that S_B is composed of columns of $[I_{n_y - m_1 + m_3}, D^T B^{-T}]$), it follows that $\forall j \in \mathbb{I}^+_{m_3}$, $(S_B)_{i,j} = 0$ if and only if i_1) and ii_1) below hold:

i_1) $(I_D)_i \leq n_y + m_2 + m_3$, which follows from the fact that the ith column of the identity matrix in (3.20) (e_i) is *not* a column of S_B if and only if $c_{D,i} = 0$, and $c_{D,i} = 0$ if and only if $(I_D)_i \leq n_y + m_2 + m_3$.

ii_1) $\forall j \in \mathbb{I}^+_{m_1+m_2+m_3}$, $(I_B)_j > n_y + m_2 + m_3$, $(D^T B^{-T})_{i,j} = 0$.

In the following it is shown that since $(S_B\tilde{c}^*)_i = 0$ if and only if i_1) and ii_1) holds, then all alternative optimal solutions (i.e. other than $x^*(p)$) to (3.11) also have the property that the $z(p)$-part of these solutions is equal to $z^o(p)$: Consider the associated PMOLP in (3.19), and let R_D be the relative cost matrix corresponding to B. Recall that, for the given p, B is a preemptive optimal basis to (3.19), and that the ith row of R_D is the relative cost vector corresponding to the cost function $\sum_{j=1}^{n_y+m_2+2m_3} W_{i,j}x_j$. Combining this with the above cited result from (Murty, 1983), it is straightforward to establish that the set of all preemptive optimal solutions to (3.19) consists of all feasible solutions to (3.19) in which the nonbasic variables $x_i(p)$ corresponding to $(R_D)_i \neq 0_{m_3}$ are equal to zero. Since $W = [0_{m_3\times(n_y+m_2+m_3)}, I_{m_3}]$, the ith column of $R_D = W_D - W_B B^{-1} D$ is equal to 0_{m_3} if and only if i_2) and ii_2) below hold:

i_2) $(W_D)_i = 0_{m_3}$. Note that $(W_D)_i = 0_{m_3}$ if and only if $(I_D)_i \leq n_y + m_2 + m_3$.

ii_2) $\forall j \in \mathbb{I}_{m_3}^+$, $(W_B B^{-1} D)_{j,i} = 0_{m_3}$. Since each column of W_B are either equal to 0_{m_3} or equal to e_k, $k \in \mathbb{I}_{m_3}^+$, and since there does not exist two non-zero columns of W_B which are equal, then $\forall i \in \mathbb{I}_{m_1+m_2+m_3}^+, \exists k \in \mathbb{I}_{m_3}^+$ such that $(W_B)_i = e_k$ implies that $\forall j \in \mathbb{I}_{n_y-m_1+m_3}^+$, $(W_B B^{-1} D)_{k,j} = (B^{-1}D)_{i,j}$, and if such a k does not exist, $\forall j \in \mathbb{I}_{n_y-m_1+m_3}^+$, $(W_B B^{-1} D)_{k,j} = 0$. Note that $(W_B)_i = e_k$ if and only if $(I_B)_i > n_y + m_2 + m_3$. Thus, $\forall j \in \mathbb{I}_{m_3}^+$, $(W_B B^{-1} D)_{j,i} = 0$ if and only if $\forall j \in \mathbb{I}_{m_1+m_2+m_3}^+$, $(I_B)_j > n_y + m_2 + m_3$, $(D^T B^{-T})_{i,j} = 0$.

At a first glance, one might believe that i_2) is not a necessary condition to ensure $(R_D)_i = 0_{m_3}$. However, this is the case due to the following: We have that $(R_D)_i = (W_D)_i - (W_B B^{-1} D)_i$. Assume that $(W_D)_i \neq 0_{m_3}$. Then $\exists k \in \mathbb{I}_{m_3}^+$, $(W_D)_i = e_k$, i.e. $(W_D)_{k,i} = 1$. This implies that $\nexists j \neq i$, $(W_B)_j = e_k$, i.e. the kth row of W_B is equal to $0_{n_y+m_2+2m_3}$. This implies that $\forall i \in \mathbb{I}_{n_y-m_1+m_3}^+$, $(W_B B^{-1} D)_{k,i} = 0$. Thus, if $(W_D)_i \neq 0_{m_3}$, $\exists k$, $(W_D)_{k,i} - (W_B B^{-1} D)_{k,i} = 1 \Leftrightarrow (R_D)_i \neq 0_{m_3}$.

Observe that i_1)$\Leftrightarrow i_2$) and ii_1)$\Leftrightarrow ii_2$). Thus, since, if $\tilde{c} = \tilde{c}^*$, it holds that $(r_D)_i = (S_B \tilde{c})_i = 0$ if and only if the corresponding $(R_D)_i = 0_{m_3}$, we have shown that for the given p, the set of all optimal solutions to (3.11) is equal to the set of all preemptive optimal solutions to (3.19). Since p was arbitrary, the lemma is proved. □

3.A.2 Proof of Lemma 3.5

First the only if-part is proved by proving the contrapositive. First, note that if B is not feasible, it is not a preemptive optimal basis. Assume that there exists a column of R_D which is lexicographically negative, i.e. assume that there exists $i \in \mathbb{I}_{m_3}^+$ and $j \in \mathbb{I}_{n_y-m_1+m_3}^+$ such that $(R_D)_{i,j} < 0$ and $\forall k \in \mathbb{I}_{i-1}^+$, $(R_D)_{k,j} = 0$. Then, introducing the jth nonbasic variable into the basis will reduce the value of the ith element of the cost vector in (3.19), while the element 1 to $i-1$ of the cost vector remain unchanged. Thus, B cannot be a preemptive optimal basis.

Next, the if-part is proved: Assume that each column of the relative cost matrix R_D corresponding to a feasible basis B to (3.19) is lexicographically nonnegative. Furthermore, assume that the ith nonbasic variable enters the basis. Since $(R_D)_i$, the ith column of R_D, is lexicographically nonnegative, the elements of Wx corresponding to the leading zeros on $(R_D)_i$ remains unchanged after the basis exchange. However, the first nonzero element in $(R_D)_i$, say the kth element, is positive, and thus the kth element of Wx increases due to this basis exchange, and thus the original cost vector is lexicographically less than the cost vector obtained after the basis exchange. □

3.A.3 Proof of Theorem 3.3

The outline of the proof is as follows: First, the claim in Step 1 in Algorithm 3.4.1 is proved, then it is proved that each column of relative cost matrix corresponding the new basis are lexicographically nonnegative, and lastly we prove that this basis is also feasible. Then, according to Lemma 3.5, the new basis is optimal to (3.19) when $p = p'$.

If $\forall j \in \mathbb{I}^+_{n_y+m_2+2m_3}$, $y_{r,j} \geq 0$, the dual problem to the single-objective LP corresponding to the first objective in (3.19), i.e. the dual of $\min \sum_{j=1}^{n_y+m_2+2m_3} W_{1,j}x$ subject to $x \in X(p)$, is unbounded (Luenberger, 1989, p. 98), and thus the primal problem is infeasible. This implies that (3.19) is infeasible when $p = p'$. This proves the claim in Step 1.

Next, we prove that the basis resulting from Algorithm 3.4.1 is a preemptive optimal basis to (3.19) when $p = p'$: Assume that $\exists j \in \mathbb{I}^+_{n_y+m_2+2m_3}$, $y_{r,j} < 0$, and for simplicity assume that $A = [B, D]$. Note that due to this assumption, the definition of $z_{i,j}$ (see Step 2 in Algorithm 3.4.1), and Lemma 3.5, we have the following relation: Let $\bar{R}_D := [0_{m_3\times(m_1+m_2+m_3)}, -R_D]$ (note the minus sign), then $z_{i,j} - c_{i,j} = (\bar{R}_D)_{i,j}$. Thus, since B is a preemptive optimal basis to (3.19) when $p \in \mathcal{P}_B$, each column of $\bar{R}_D$ is lexicographically nonpositive (LNP). Let B' be the basis obtained by replacing the rth column of B by the qth column of A, and let $z'_{i,j} := (c^i_B)'(B')^{-1}A_j$ and $(\bar{R}'_D)_{i,j} := (z_{i,j} - c_{i,j})' := z'_{i,j} - c_{i,j}$, where $(c^i_B)'$ consists of the elements of the ith row of W corresponding to the new basis B'. Then, by pivoting on element $y_{r,q}$ in the tableau corresponding to B, the we obtain

$$(z_{i,j} - c_{i,j})' = (z_{i,j} - c_{i,j}) - \frac{y_{r,j}}{y_{r,q}}(z_{i,q} - c_{i,q}),\ i \in \mathbb{I}^+_{m_3},\ j \in \mathbb{I}^+_{n_y+m_2+2m_3}. \tag{3.28}$$

In the following it will be shown that each column of $\bar{R}'_D$ is LNP, and we prove it separately for columns j of $\bar{R}'_D$ corresponding to $y_{r,j} < 0$, $y_{r,j} = 0$, and $y_{r,j} > 0$ respectively.

First, consider the columns of $\bar{R}'_D$ corresponding to $y_{r,j} < 0$, i.e. column $j \in J_0$. Due to the way r and q are computed by Algorithm 3.4.1, $\forall j \in J_0, \exists k_j \in \mathbb{I}_{m_3}$ such that

$$\left(\left(\forall i \in \mathbb{I}^+_{m_3}, i \le k_j, \frac{z_{i,j} - c_{i,j}}{y_{r,j}} = \frac{z_{i,q} - c_{i,q}}{y_{r,q}}\right) \text{ and } \left(k_j < m_3 \Rightarrow \frac{z_{k_j+1,j} - c_{k_j+1,j}}{y_{r,j}} > \frac{z_{k_j+1,q} - c_{k_j+1,q}}{y_{r,q}}\right)\right)$$

By (3.28), we have (recall that $y_{r,q} < 0$):

$$\forall i \in \mathbb{I}^+_{m_3}, i \le k_j : \frac{z_{i,j} - c_{i,j}}{y_{r,j}} = \frac{z_{i,q} - c_{i,q}}{y_{r,q}} \Rightarrow (z_{i,j} - c_{i,j})' = 0$$
$$i = k_j + 1 : \frac{z_{i,j} - c_{i,j}}{y_{r,j}} > \frac{z_{i,q} - c_{i,q}}{y_{r,q}} \Rightarrow (z_{i,j} - c_{i,j})' < 0$$

Thus, $\forall j \in J_0$, $((z_{1,j} - c_{1,j})', \dots, (z_{m_3,j} - c_{m_3,j})')$ is LNP.

Next, consider the columns j of $\bar{R}'_D$ corresponding to $y_{r,j} = 0$: By (3.28), we have (recall that $\forall j \in \mathbb{I}^+_{n_y+m_2+2m_3}$, $((z_{1,j} - c_{1,j}), \dots, (z_{m_3,j} - c_{m_3,j}))$ is LNP): $\forall i \in \mathbb{I}^+_{m_3}$ $(z_{i,j} - c_{i,j})' = (z_{i,j} - c_{i,j})$, and thus $\forall j \in \{j \in \mathbb{I}^+_{n_y+m_2+2m_3} \,|\, y_{r,j} = 0\}$, $((z_{1,j} - c_{1,j})', \dots, (z_{m_3,j} - c_{m_3,j})')$ is LNP.

Finally, consider the columns j of $\bar{R}'_D$ corresponding to $y_{r,j} > 0$: We have the following three cases: *i)* $(z_{1,j} - c_{1,j}) < 0$, *ii)* $(z_{1,j} - c_{1,j}) = 0$ and $(z_{1,q} - c_{1,q}) < 0$, and *iii)* $(z_{1,j} - c_{1,j}) = 0$ and $(z_{1,q} - c_{1,q}) = 0$. Since $\frac{y_{r,j}}{y_{r,q}} < 0$ and $(z_{1,q} - c_{1,q}) \le 0$, it is easily seen from (3.28) that for case *i)* and *ii)*, $(z_{1,j} - c_{1,j})' < 0$, and thus $((z_{1,j} - c_{1,j})', \dots, (z_{m_3,j} - c_{m_3,j})')$ is LNP. For case *iii)*, obviously, $(z_{1,j} - c_{1,j})' = 0$. Thus, to establish that $((z_{1,j} - c_{1,j})', \dots, (z_{m_3,j} - c_{m_3,j})')$ is LNP, it is necessary to consider $(z_{2,j} - c_{2,j})'$. For this case, there are also three cases to consider, which are similar to case *i)* to *iii)*. Continuing this way, for a $j \in \{j \in \mathbb{I}^+_{n_y+m_2+2m_3} \,|\, y_{r,j} > 0\}$, we get the following: $\forall i \in \mathbb{I}^+_{m_3}$, $(z_{i,j} - c_{i,j})' = 0$ if and only if $\forall i \in \mathbb{I}^+_{m_3}$, $(z_{i,p} - c_{i,p})' = (z_{i,j} - c_{i,j})' = 0$. For all other $j \in \{j \in \mathbb{I}^+_{n_y+m_2+2m_3} \,|\, y_{r,j} > 0\}$, $((z_{1,j} - c_{1,j})', \dots, (z_{m_3,j} - c_{m_3,j})')$ is lexicographically negative. Thus, since each column $\bar{R}'_D$ are LNP, each column of the corresponding relative cost matrix are LNN.

Finally, we show that B' is a feasible basis to (3.19) when $p = p'$, i.e. that $x'_B(p') := (B')^{-1}b(p') \ge 0$: By pivoting on element $y_{r,q}$ in the tableau corresponding to B, the we obtain

$$(x'_B(p'))_i = (x_B(p'))_i - \frac{(x_B(p'))_r}{y_{r,q}} y_{i,q}, \; i \in \mathbb{I}^+_{m_1+m_2+m_3} \setminus r \tag{3.29}$$

$$(x'_B(p'))_r = \frac{(x_B(p'))_r}{y_{r,q}} \tag{3.30}$$

By choosing ε small enough, $(x_B(p'))_r < 0$ can be made arbitrarily close to zero, and thus by (3.29), $(x'_B(p'))_i > 0$, $i \in \mathbb{I}^+_{m_1+m_2+m_3} \setminus r$. Further, since $(x_B(p'))_r < 0$ and $y_{r,q} < 0$, by (3.30), $(x'_B(p'))_r > 0$. Thus B' is a feasible basis to (3.19) when $p = p'$. Thus, since, in addition, each column of the corresponding relative cost matrix is lexicographically nonnegative, according to Lemma 3.5, B' is a preemptive optimal basis to (3.19) when $p = p'$. □

3.B A counter example

As described at the end of Section 3.4.2, for the single-objective case, Gal (1995) proposes the following method to compute a $\tilde{\mathcal{B}}$ such that $\forall B', B'' \in \tilde{\mathcal{B}}$, $B' \neq B''$, $\mathcal{P}_{B'}$ and $\mathcal{P}_{B''}$ are non-overlapping and such that $\mathcal{P} \subseteq \cup_{B \in \tilde{\mathcal{B}}} \mathcal{P}_B$: When, in the single objective version of Algorithm 3.4.1, there are several possible neighboring bases along the same facet of a $\mathcal{P}_{B^k}$ corresponding to a given basis index set ρ^k, arbitrarily select one of them to enter U_k in Step 2 in Algorithm 3.4.2. This section contains an explanation of why this strategy may fail. Consider the imaginary example depicted in Figure 3.3. In the figure, each polytope corresponds to a region $\mathcal{P}_{B^k}$, where B^k is an optimal[3] basis, and the integers denote the value of the iteration counter k in Algorithm 3.4.2 when this basis was added to set T_k. Thus, the upper left polytope in Figure 3.3 illustrates the area where the optimal basis B^0 computed in Step 1 in Algorithm 3.4.2 is feasible. Assume that, in Step 2 in Algorithm 3.4.2, ρ^k (i.e. basis B^k) are selected from U_{k-1} in the order illustrated in Figure 3.3. According to the figure, the bases corresponding to polytope $0a$ and $0b$ (i.e. B^{0a} and B^{0b}) are both neighbors to B^0 along the lower facet of $\mathcal{P}_{B^0}$. At $k = 0$, assume that ρ^{0a} is added to U_0 (along with a set of neighbors along the other facets of $\mathcal{P}_{B^0}$.) Thus, ρ^{0b} is not considered further by the algorithm. Further, there are two neighboring bases along the upper facet of $\mathcal{P}_{B^5}$: B^{5a} and B^{5b}. At $k = 5$, assume that B^{5b} is selected to enter U_5. According to the figure, along the upper facet

[3] Note that we write optimal instead of preemptive optimal. This is because we here consider a single-objective parametric LP such as (3.11).

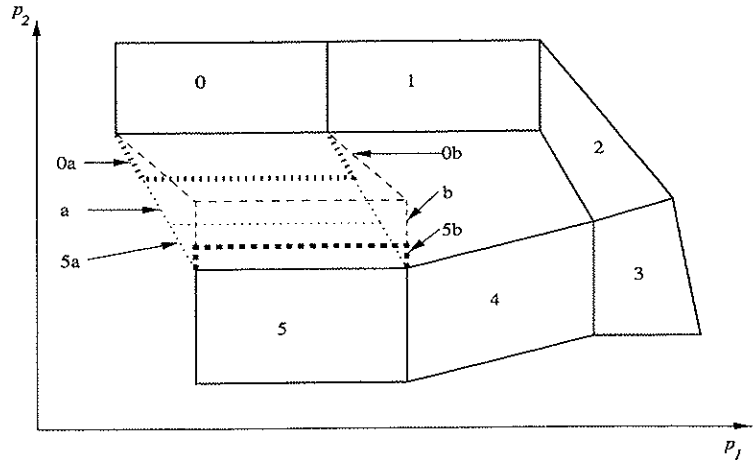

Figure 3.3: An imaginary example illustrating why Algorithm 3.4.2 may fail.

of $\mathcal{P}_{B^{5b}}$, there is a single neighbor B^b which overlaps $\mathcal{P}_{B^a}$, where B^a is the only neighbor along the lower facet of $\mathcal{P}_{B^{0a}}$. Thus, in order to cover $\mathcal{P}$ when these selections are made, overlapping regions are unavoidable. Note that, if, at $k = 5$, B^{5a} has been selected as the neighbor along the upper facet of $\mathcal{P}_{B^5}$ instead of B^{5b}, this situation would not have appeared. Also note that at $k = 5$, there is no information available which could have been used to determine the selection in order to guarantee non-overlapping regions. However, if $\mathcal{B}$, the set of all optimal bases were available, this situation could have been avoided by using the theory in (Gal, 1995, Chapter 4). Note that the modification proposed at the end of Section 3.4.2 solves this problem without the need of calculating $\mathcal{B}$. Although this example is imaginary, it illustrates the main features of more complex examples encountered when applying this algorithm on the MPC example in Section 3.5.

3.C Computing ratios

For completeness reasons, in this section we present how the ratios described in Section 3.4.4 can be computed in a manner such that the ratios obtained by using this strategy generally becomes less than the ratios computed by using the strategy proposed in (Vada, Slupphaug and Johansen, 1999a). Note that small ratios is desired in order to avoid numerical problems when solving the LP problem (3.6).

Before presenting the strategy, we present an example which shows that it is not sufficient to find ratios $r_i(p)$, $i \in \mathbb{I}^+_{m_3-1}$, such that the corresponding weights solve the OWDP for this particular p, and then maximize *these* ratios over $\mathcal{P}$:

Example 3.1
Assume that

$$G^1 = \begin{bmatrix} 5 & -10 & 10 \end{bmatrix} \qquad g^1(p) = 16$$

$$G^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad g^2(p) = \begin{bmatrix} 2 & 2 & 2 \end{bmatrix}$$

$$G^3 = \begin{bmatrix} 5 & 7 & 6 \\ -9 & -14 & 3 \\ 19 & -23 & 19 \\ -6 & 13 & -2 \end{bmatrix} \qquad g^3(p) = \begin{bmatrix} 0 & -20 & 15 & 3 \end{bmatrix}.$$

Note that for simplicity, $g^i(p)$ are independent of p. The corresponding A and b in (3.11), becomes

$$A = \begin{bmatrix} 5 & -10 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 7 & 6 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -9 & -14 & 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 19 & -23 & 19 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ -6 & 13 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}, \qquad b = \begin{bmatrix} 16 \\ 2 \\ 2 \\ 2 \\ 0 \\ -20 \\ 15 \\ -3 \end{bmatrix}.$$

The optimal basis variables are $x_3, x_4, x_5, x_6, x_{10}, x_{11}, x_{12}, x_{13}$, and the corresponding S_B becomes

$$S_B = \begin{bmatrix} 2 & -10.5 & 9.5 & 0 \\ 13 & -11 & -4 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Assume that $\tilde{c} = (\frac{15}{13}, 1, 1, 1)$ (i.e. $r = (\frac{15}{13}, 1, 1)$). Then $S_B\tilde{c} \geq 0$, and $\tilde{c}$ solves the OWDP in this case. Define $\tilde{c}'$ where $\tilde{c}'_i := k^{(m_3-i)}\frac{\tilde{c}_i}{\tilde{c}_{i+1}}$, $i =$

$1, \ldots, 3$. Then $S_B\tilde{c}' < 0$ if $k \in (1.246, 3.304)$. E.g., if $k = 2$, $\tilde{c}' = (\frac{120}{13}, 4, 2, 1)$, and $S_B\tilde{c}' = (-4.5385, 68.0, 9.2308, 4, 2, 1)$. Now, imagine that b is a function of p and assume that there is a $k \in (1.246, 3.304)$ such that $\tilde{c}'$ solves the OWDP for a $p' \neq p$. Then, the weights associated with the ratios obtained by maximizing the ratios solving the OWDP for p and p' individually will not solve the OWDP for both p and p', showing that such a maximization generally will not work. (Note that if $\tilde{c} = (5.25, 1, 1, 1)$, then $S\tilde{c}' \geq 0$ $\forall k \geq 1$.) ■

The point is that the ratios have to be sufficiently large to guarantee that such a maximization will work.

For a fixed $p \in \mathcal{P}$, we now derive ratios $r_i^*(p)$ such that $\forall \tilde{c} \in \tilde{C}(r^*(p))$, $x(p)$ is optimal to (3.11) if $z(p) = z^o(p)$. Then, after presenting the algorithm, Lemma 3.6 states that this implication can also be reversed. Let B be a preemptive optimal basis to (3.19) and let $x^*(p)$ be the corresponding (optimal) basic solution. Since B is a feasible basis to (3.11), $\forall \tilde{c} \in \tilde{C}(r^*(p))$, $S_B\tilde{c} \geq 0$ (i.e. positive relative cost vector associated with B) implies that $\forall \tilde{c} \in \tilde{C}(r^*(p))$, $x^*(p)$ is an optimal solution to (3.11) (Luenberger, 1989, p. 43).

Assume that $\exists i \in \mathbb{I}^+_{n_y-m_1+m_3} \forall j \in \mathbb{I}^+_{m_3}, (S_B)_{i,j} = 0$. Then $\forall \tilde{c} \in \mathbb{R}^{m_3}$, $(S_B\tilde{c})_i = 0$. Next, assume that $\exists i \in \mathbb{I}^+_{n_y-m_1+m_3} \forall j \in \mathbb{I}^+_{m_3-1}, (S_B)_{i,j} = 0$ $\wedge$ $(S_B)_{i,m_3} \neq 0$. Then, due to Lemma 3.3, $(S_B)_{i,m_3} > 0$, and $\forall r^*(p) \geq 0, \tilde{c} \in \tilde{C}(r^*(p))$, $(S_B\tilde{c})_i \geq 0$. Thus, for simplicity, in the following assume that each row of S_B has less than $m_3 - 1$ leading zeros. Let $\zeta \leq m_3 - 2$ be the index to the first nonzero element of the ith row of S_B. Due to Lemma 3.3, $(S_B)_{i,\zeta} > 0$. Note that $\tilde{c} \in \tilde{C}(r^*(p))$ is equivalent to $\exists \omega_k > 1$, $k \in \mathbb{I}^+_{m_3-1}$ such that $\tilde{c}_i = \prod_{k=i}^{m_3-1} r_k^*(p)\omega_k\tilde{c}_{m_3}$, $i \in \mathbb{I}^+_{m_3-1}$. Optimality of x $\forall \tilde{c} \in \tilde{C}(r^*(p))$ is equivalent to $\forall \tilde{c} \in \tilde{C}(r^*(p))$, $i \in \mathbb{I}^+_{n_y-m_1+m_3}$, $(S_B\tilde{c})_i \geq 0$, i.e. $\forall i \in \mathbb{I}^+_{n_y-m_1+m_3}$, $s \in \{\zeta, \ldots, m_3 - 1\}$, $\omega_s > 1$,

$$\begin{aligned}(S_B\tilde{c})_i =& (S_B)_{i,\zeta}\tilde{c}_{m_3} \prod_{s=\zeta}^{m_3-1} \omega_s r_s^*(p) + \cdots + (S_B)_{i,\zeta+j}\tilde{c}_{m_3} \prod_{s=\zeta+j}^{m_3-1} \omega_s r_s^*(p) + \\ & \cdots + (S_B)_{i,m_3}\tilde{c}_{m_3} \geq 0\end{aligned}$$

$$\begin{aligned}\Leftrightarrow r_\zeta^*(p) \geq -\frac{1}{(S_B)_{i,\zeta}\omega_\zeta \prod_{s=\zeta+1}^{m_3-1} \omega_s r_s^*(p)} & \left((S_B)_{i,\zeta+1} \prod_{s=\zeta+1}^{m_3-1} \omega_s r_s^*(p) + \cdots \right. \\ & \left. (S_B)_{i,\zeta}\omega_\zeta \prod_{s=\zeta+1}^{m_3-1} \omega_s r_s^*(p) \right) \quad (3.31)\end{aligned}$$

Define $\sigma_i^\zeta(\omega_\zeta, \dots, \omega_{m_3-1}; r^*_{\zeta+1}(p), \dots, r^*_{m_3-1}(p))$ as the right hand side of (3.31), i.e.

$$\sigma_i^\zeta(\omega_\zeta, \dots, \omega_{m_3-1}; r^*_{\zeta+1}(p), \dots, r^*_{m_3-1}(p)) := -\frac{1}{(S_B)_{i,\zeta}\omega_\zeta}\Bigg((S_B)_{i,\zeta+1} + \cdots + (S_B)_{i,\zeta+j}\left(\prod_{s=\zeta+1}^{\zeta+j-1}\omega_s r^*_s(p)\right)^{-1} + \cdots + (S_B)_{i,m_3}\left(\prod_{s=\zeta+1}^{m_3-1}\omega_s r^*_s(p)\right)^{-1}\Bigg) \quad (3.32)$$

$$= \frac{\alpha_{i,\zeta}}{\omega_\zeta} + \cdots + \frac{\alpha_{i,\zeta+j}}{\omega_\zeta\omega_{\zeta+1}\cdots\omega_{\zeta+j}} + \cdots + \frac{\alpha_{i,m_3-1}}{\omega_\zeta\omega_{\zeta+1}\cdots\omega_{m_3-1}}, \quad (3.33)$$

where $\alpha_{i,j} := \frac{-(S_B)_{i,j+1}}{(S_B)_{i,\zeta}}\left(\prod_{s=\zeta+1}^{j} r^*_s(p)\right)^{-1}$, $j = \zeta, \dots, m_3-1$. Let $\mathcal{I}_\zeta$ denote the set of indices to rows of S_B having the ζth element as the first nonzero element, and let $\mathcal{J} := \{i \in \mathbb{I}^+_{m_3-1} \mid \exists j \in \mathbb{I}^+_{n_y-m_1+m_3}$ such that i is the first nonzero element in the jth row of $S_B\}$. In order to satisfy $\forall i \in \mathbb{I}^+_{n_y-m_1+m_3}, \tilde{c} \in \tilde{C}(r^*(p))$, $(S_B\tilde{c})_i \geq 0$, a sufficiently large $r^*_\zeta(p)$ is $r^*_\zeta(p) = \tilde{r}_\zeta(p;\underline{r})$, where

$$\tilde{r}_\zeta(p;\underline{r}) = \max\left\{\underline{r}, \max_{i\in\mathcal{I}_\zeta}\hat{\sigma}_i^\zeta(\tilde{r}_{\zeta+1}(p;\underline{r}), \dots, \tilde{r}_{m_3-1}(p;\underline{r}))\right\}, \ \forall\zeta \in \mathcal{J}, \quad (3.34)$$

where $\underline{r} > 0 \in \mathbb{R}$, and where

$$\hat{\sigma}_i^\zeta(\tilde{r}_{\zeta+1}(p;\underline{r}), \dots, \tilde{r}_{m_3-1}(p;\underline{r})) := \sup_{\omega_l>1,\ l=\zeta,\dots,m_3-1} \sigma_i^\zeta(\omega_\zeta, \dots, \omega_{m_3-1}; \tilde{r}_{\zeta+1}(p;\underline{r}), \dots, \tilde{r}_{m_3-1}(p;\underline{r}))$$

By using the nonlinear variable transformation $\chi_{\zeta,i} := \frac{1}{\prod_{j=\zeta}^{\zeta+i}\omega_j}$, $i \in \mathbb{I}^+_{m_3-\zeta-1}$, the nonlinear programming problem of maximizing $\sigma_i^\zeta(\omega_\zeta, \dots, \omega_{m_3-1}; \tilde{r}_{\zeta+1}(p;\underline{r}), \dots, \tilde{r}_{m_3-1}(p;\underline{r}))$ with respect to ω_i, $i = \zeta, \dots, m_3 - 1$, is equivalent to the following LP problem:

$$\hat{\sigma}_i^\zeta(\tilde{r}_{\zeta+1}(p;\underline{r}), \dots, \tilde{r}_{m_3-1}(p;\underline{r})) = \max_{0\leq\chi_{\zeta,1}\leq 1,\ 0\leq\chi_{\zeta,i}\leq\chi_{\zeta,i-1},\ i=2,\dots,m_3-\zeta-1} \alpha_{i,\zeta}\chi_{\zeta,1} + \cdots + \alpha_{i,m_3-1}\chi_{\zeta,m_3-\zeta-1}. \quad (3.35)$$

$\hat{\sigma}_i^\zeta(\tilde{r}_{\zeta+1}(p;\underline{r}), \dots, \tilde{r}_{m_3-1}(p;\underline{r}))$ is bounded, since $\chi_{\zeta,i}$, $i \in \mathbb{I}^+_{m_3-\zeta-1}$ is bounded, and $\alpha_{i,j}$, $j = \zeta, \dots, m_3 - 1$ have finite values. (Note that there does not necessarily exist ω_i, $i = \zeta, \dots m_3$, such that $\sigma_i^\zeta(\cdot;\cdot) = \hat{\sigma}_i^\zeta(\cdot;\cdot)$. However, in the problem considered here, only the scalar $\hat{\sigma}_i^\zeta(\cdot;\cdot)$ is needed.) Note that $\tilde{r}_\zeta(p;\underline{r})$ in (3.34) is parameterized by $\tilde{r}_i(p;\underline{r})$, $i \in \{\zeta+1, \dots, m_3-1\}$ and $\underline{r}$. In order to compute $r^*_i(p)$, $i \in \mathbb{I}^+_{m_3-1}$, the following algorithm may be used:

Algorithm 3.C.1
Given a preemptive optimal basis B to (3.19).

Step1: Set $i = m_3 - 1$, and fix $\underline{r} > 0$.

Step 2: If $i \in \mathcal{J}$, compute $\tilde{r}_i(p;\underline{r})$ by solving (3.34) and (3.35), else, set $\tilde{r}_i(p;\underline{r}) = \underline{r}$.

Step 3: If $i > 1$, set $i \leftarrow i - 1$, and go to Step 2, else set $r^*(p) = \tilde{r}(p;\underline{r})$ stop.

LEMMA 3.6
Assume A1 and A2, and assume that $p \in \mathcal{P}$ is fixed. Then the ratios $r^*(p)$ computed by Algorithm 3.C.1 has the following property: $\forall \tilde{c} \in \tilde{C}(r^*(p))$, $x(p) = (y(p), v(p), w(p), z(p))$ is an optimal solution to (3.11) if and only if *$z(p) = z^o(p)$ and $x(p)$ is a feasible solution to (3.11).*

Proof: The proof follows the same line as the proof of Lemma 3.4. □
Given a set of bases $\mathcal{B}$ as defined in Section 3.4. Due to the definition of $\tilde{C}(r)$ (see (3.15)), an $\hat{r}^*$ such that $\forall \tilde{c} \in \tilde{C}(\hat{r}^*)$ $\forall p \in \cup_{B \in \mathcal{B}} \mathcal{P}_B$ $x(p)$ is an optimal solution to (3.11) *if and only if* $z(p) = z^o(p)$ (i.e. if and only if $x(p)$ is a preemptive optimal solution to (3.19)) can be calculated as follows

$$\hat{r}_i^* = \max_{B \in \mathcal{B}} (r_B^*)_i, \quad i \in \mathbb{I}_{m_3-1}^+ \tag{3.36}$$

where r_B^* is computed by using Algorithm 3.C.1.

3.D Influence of $\underline{r}$ in Algorithm 3.C.1

If $r^*(p)$ are computed by Algorithm 3.C.1, the scalar parameter $\underline{r}$ determines the least value of any ratio $r_i^*(p)$, i.e. $\forall i \in \mathbb{I}_{m_3-1}^+$, $r_i^*(p) \geq \underline{r}$. (Note that in Algorithm 3.C.1, $\tilde{r}_i(p;\underline{r})$ is parameterized by $\underline{r}$ and $\tilde{r}_j(p;\underline{r})$, $j \in \{i+1, \dots, m_3 - 1\}$). Since numerical problems when solving (3.11) may occur if the ratio between the greatest and least element of $\tilde{c} \in \tilde{C}(\hat{r}^*)$ becomes very large, $\underline{r}$ should be set to a value such that the resulting $\tilde{C}(\hat{r}^*)$ contains $\tilde{c}$s do that not cause such numerical problems. In the following, the influence of $\underline{r}$ on $\tilde{r}_i(p;\underline{r})$ is discussed. Let

$$\alpha'_{i,j} := \frac{-(S_B)_{i,j+1}}{(S_B)_{i,\zeta}}, \; j = \zeta, \dots, m_3 - 1.$$

Then (3.32) can be reformulated as (recall that $\chi_{\zeta,i} := \frac{1}{\prod_{j=\zeta}^{\zeta+i} \omega_j}$)

$$\sigma_i^\zeta(\cdot;\cdot) = \sum_{j=\zeta}^{m_3-1} \frac{\alpha'_{i,\zeta}}{\prod_{k=\zeta+1}^{j} r_k^*(p)} \chi_{\zeta,j}.$$

Note that, due to (3.34), $r_k^*(p)$ in (3.37) (and in the rest of this section) is dependent on $\underline{r}$. However, for notational ease, this dependency is not explicitly shown. Assume that one or more $r_i^*(p) = \underline{r}$, $i \in \{\zeta+1, \ldots, m_3-1\}$, and let $\mathcal{I}_j^{\underline{r}} := \{i \in \{j, \ldots, m_3-1\} \mid r_i^*(p) = \underline{r}\}$. Then (3.35) can be reformulated as

$$\hat{\sigma}_i^\zeta(\cdot,\cdot) = \max_{0 \le \chi_{\zeta,1} \le 1,\ 0 \le \chi_{\zeta,i} \le \chi_{\zeta,i-1},\ i=2,\ldots,m_3-\zeta-1} \sum_{j=\zeta}^{m_3-1} \frac{\alpha'_{i,\zeta}\underline{r}^{-|\mathcal{I}_j^{\underline{r}}|}}{\prod_{k\in\{\zeta+1,\ldots,m_3-1\}\setminus\mathcal{I}_j^{\underline{r}}} r_k^*(p)} \chi_{\zeta,j}. \quad (3.37)$$

The constraints in (3.37) have the following $m_3-\zeta-1$ extreme points: $\chi_\zeta^1 = (1,\ldots,1)$, $\chi_\zeta^2 = (1,\ldots,1,0), \cdots, \chi_\zeta^{m_3-\zeta-2} = (1,0,\ldots,0), \chi_\zeta^{m_3-\zeta-1} = (0,\ldots,0)$. In other words, there exists a $q \in \mathbb{I}_{m_3-\zeta-1}^+$ such that χ_ζ^*, the optimum to (3.37), is equal to χ_ζ^q, that is

$$\hat{\sigma}_i^\zeta(\cdot,\cdot) = \sum_{j=\zeta}^{q} \frac{\alpha'_{i,\zeta}\underline{r}^{-|\mathcal{I}_j^{\underline{r}}|}}{\prod_{k\in\{\zeta,\ldots,m_3-1\}\setminus\mathcal{I}_j^{\underline{r}}} r_k^*(p)}. \quad (3.38)$$

(3.38), (3.36), and (3.34) show how $\underline{r}$ influences on $\hat{r}_i^*$. From a numerical point of view, $\underline{r}$ should be chosen so as to obtain a $\tilde{C}(\hat{r}^*)$ which contains $\tilde{c}$s having the property that the largest ratio $\frac{\tilde{c}_i}{\tilde{c}_j}$, $i,j \in \mathbb{I}_{m_3}^+$ is small. Since $\tilde{c} \in \tilde{C}(\hat{r}^*) \Leftrightarrow \frac{\tilde{c}_i}{\tilde{c}_{i+1}} > \hat{r}_i^*$, and thus $\forall \tilde{c} \in \tilde{C}(\hat{r}^*)$, $i,j \in \mathbb{I}_{m_3-1}^+$, $i<j$, $\frac{\tilde{c}_i}{\tilde{c}_j} > \prod_{k=i}^{j-1} \hat{r}_k^*$, it follows that

$$\inf_{\tilde{c}\in\tilde{C}(\hat{r}^*)} \max_{j>i \wedge i,j\in\mathbb{I}_{m_3-1}^+} \frac{\tilde{c}_i}{\tilde{c}_j} = \max_{i,j\in\mathbb{I}_{m_3-1}^+} \prod_{k=i}^{j-1} \hat{r}_k^* \quad (3.39)$$

and

$$\inf_{\tilde{c}\in\tilde{C}(\hat{r}^*)} \max_{j<i \wedge i,j\in\mathbb{I}_{m_3-1}^+} \frac{\tilde{c}_i}{\tilde{c}_j} = \max_{i,j\in\mathbb{I}_{m_3-1}^+} \prod_{k=i}^{j-1} (\hat{r}_k^*)^{-1}. \quad (3.40)$$

Thus, $\underline{r}$ should be selected as a solution to $\min_{\underline{r}} r^{\max}(\underline{r})$ where

$$r^{\max}(\underline{r}) := \max\left\{ \max_{i,j\in\mathbb{I}_{m_3-1}^+} \prod_{k=i}^{j-1} \hat{r}_k^*,\ \max_{i,j\in\mathbb{I}_{m_3-1}^+} \prod_{k=i-1}^{j} (\hat{r}_k^*)^{-1} \right\}, \quad (3.41)$$

(3.41) is generally a non-convex optimization problem, which, in general, is very hard to solve. The non-convex nature has been established empirically. Moreover, the number of functions to evaluate in order to solve (3.41) may become large in real problems due to the following: There may be a large number of bases, and for each S_B corresponding to a basis B, there can be up to $n_y - m_1 + m_3$ different LP problems (3.35) which need to be solved in order to compute $\tilde{r}_i(p, \underline{r})$, $i \in \mathbb{I}^+_{m_3-1}$ (see (3.34)), and in (3.41), there are $6m_3 - 6$ functions to maximize. To conclude, if $r^{\max}(\underline{r})$ becomes prohibitively large, it might be of help to change $\underline{r}$, and then compute $r^*(p)$ again to check whether or not the value of (3.41) is less.

Chapter 4

Application, Computational Issues and Stability

This chapter is a reprint of (Vada et al., 2000), which was submitted to Automatica in January 2000. Parts of the results in this paper are also also given in (Vada, Slupphaug, Johansen and Foss, 1999).

Abstract In order to minimize the number of situations when a model predictive controller (MPC) fails to compute a control input, all practical MPC implementations should have a means to recover from infeasibility. We discuss several aspects related to infeasibility handling, and we present a recently developed infeasibility handler which computes optimal relaxations of the relaxable constraints subject to a user-defined prioritization. This infeasibility handler requires that only *a single* linear program needs to be solved on-line in addition to the standard quadratic programming problem. A stability result for this infeasibility handler combined with the Rawlings-Muske MPC controller is provided, and various practical and computational issues are discussed. The method is illustrated on a simulated FCCU main fractionator, and from the results we conclude that the proposed strategy for designing the proposed infeasibility handler is applicable on problems of realistic size.

Keywords: Model based control, infeasibility handling, linear programming, linear systems.

4.1 Introduction

During the last years, model predictive control (MPC) has become an attractive control strategy within the process industries. Important stability results within the area of linear MPC are given in (Rawlings and

Muske, 1993) under the assumption of feasibility. In order to fully exploit this stabilizing property, a means to recover from infeasibility of the associated optimization problem whenever possible is required. Note that in the MPC controller proposed by Rawlings and Muske (1993), an approach for handling infeasibilities caused by the state constraints is included. Infeasibility problems may occur due to e.g. disturbances, operator intervention, modelling errors, or plant failures.

Constraints representing physical limitations must be enforced at all times (non-relaxable). Other constraints should be satisfied whenever possible (relaxable), but may be relaxed when necessary. In order to transform an infeasible MPC optimization problem into a feasible one, there must exist a solution to the non-relaxable constraints. If no such solution exists, some alternative control strategy must be activated. Note that in a typical MPC implementation, there is a large number of constraints. When infeasibility occurs, it is often not obvious which constraints to relax and the amount that these constraints should be relaxed in order to render a consistent set of constraints.

There exist techniques which transform an infeasible MPC-problem into a feasible one by treating equally all constraints which can be relaxed, see e.g. (Garcia and Morshedi, 1986), (Rawlings and Muske, 1993), (Qin and Badgwell, 1997) and (Scokaert and Rawlings, 1999). However, the constraints are often not equally important, e.g. it is usually more important to satisfy the safety constraints than a product quality constraint. One way to explicitly express this difference in importance is to give the constraints different priority levels, and then specify that minimizing the violation of a constraint with a given priority level is "infinitely more important" than it is to minimize the violation of any of the constraints with a lower priority level (hard prioritization). There are some existing techniques which take such prioritization levels into account when recovering from infeasibility. IDCOM-M (Setpoint Inc.), HIECON and PFC (both from Adersa) provide a means of recovering from infeasibilities which involves prioritization of the constraints. When the on-line optimization problem becomes infeasible, the lowest prioritized constraints are dropped (Qin and Badgwell, 1997).

In (Alvarez and de Prada, 1997), a heuristic infeasibility handler which treats the constraints on the control inputs and outputs in a separate manner is proposed. Several strategies are proposed in order to relax the constraints, and different approaches may be assigned to the different constraints. However, these strategies does not use priority levels when computing the constraint violations.

Scokaert (1994) proposes several strategies to solve infeasibility problems, including strategies involving hard prioritization. The most rigorous approach is to satisfy as many of the highest prioritized constraints as possible, and then compute a feasible relaxation of the other constraints by treating them as soft constraints, that is, a term is added to the cost function in the original MPC optimization problem which penalizes the violations of these constraints (soft prioritization). However, he does not discuss *how* to compute the set of constraints which can be satisfied without any violation.

Tyler and Morari (1999) presents an approach for infeasibility handling with hard prioritized constraints. In their approach, integer variables are introduced to cope optimally with the prioritization. The minimization of the size of the violation of the constraints is performed according to their prioritization by solving a sequence of optimization problems.

In (Vada, Slupphaug and Foss, 1999) an algorithm is presented which, in case of infeasibility of the MPC-problem, optimally takes the hard prioritization among the constraints into account when relaxing the constraints. This algorithm includes a sequence of linear programming (LP) or quadratic programming (QP) problems to be solved at on-line every sample. The main difference between the approach described in (Vada, Slupphaug and Foss, 1999) and the one presented in (Tyler and Morari, 1999) is that the latter approach results in a sequence of mixed integer LP (or mixed integer QP) problems in addition to the original MPC optimization problem, while the former approach results in a sequence of LP (or QP) problems in addition to the original MPC optimization problem. Note that the number of optimization problems needed to be solved in the first approach is generally less than in the latter approach. However, if the sampling time is short compared to the number and size of the optimization problems to be solved, both approaches may be prohibitively time consuming.

In (Meadowcroft et al., 1992) a modular multivariable controller (MMC) is developed, which is based on the solution of a multi-objective optimization problem using the strategy of lexicographic goal programming where the objectives have different priorities. This solution strategy implies that the optimization problem is solved sequentially, and thus suffers from the same problems related to computational time as the approaches in (Tyler and Morari, 1999) and (Vada, Slupphaug and Foss, 1999). ((Meadowcroft et al., 1992) contains a detailed methodology for the design of steady state MMCs only.)

An important difference between the algorithms presented in (Tyler and Morari, 1999), (Scokaert, 1994), (Meadowcroft et al., 1992) and (Vada, Slupphaug and Foss, 1999), and the other approaches mentioned above which *also* take prioritization into account, is that the algorithms presented in these papers minimize the violations of those constraints which cannot be fulfilled. Just dropping a set of constraints may result in unnecessary large constraint violations, and will in this sense be suboptimal.

In (Vada, Slupphaug and Johansen, 1999a), the problem of determining the optimal constraint violations subject to hard prioritization is formulated as a single LP problem, together with an existence proof and an outline of an algorithm to design this LP. The full details of the design of this LP problem are given in (Vada, Slupphaug and Johansen, 1999b). To the best of the authors knowledge, this strategy is the only optimal infeasibility handler which considers *hard prioritized* constraints without the use of a sequential, computationally expensive solution approach.

While the focus in (Vada, Slupphaug and Johansen, 1999a) and (Vada, Slupphaug and Johansen, 1999b) is on existence of the above described LP problem and on how to compute its parameters, the focus in the present paper is on the application of the proposed infeasibility handler. The usefulness of the method is illustrated on a simulated distillation column, and some practical modifications of the problem formulation in (Vada, Slupphaug and Johansen, 1999a) and (Vada, Slupphaug and Johansen, 1999b) are suggested in order to allow for several constraints sharing the same priority level. These modifications also reduce the off-line computational load or the memory requirements required to design the LP. Further, the present paper provides a novel stability result for this infeasibility handler combined with the Rawlings-Muske MPC controller (Rawlings and Muske, 1993).

The following notation is used throughout the paper: Let $n \geq 1$ be an integer and $x, y \in \mathbb{R}^n$. Then $\mathbb{I}_n^+ := \{1, \ldots, n\}$, $x \geq (>) y \Leftrightarrow x_i \geq (>) y_i$, $i \in \mathbb{I}_n^+$, and 0_n is an n-dimensional vector with zeros. (x, y) is used to express $[x^T, y^T]^T$. I_n is the $n \times n$ identity matrix, $I_0 = \emptyset$, $|J|$ denotes the cardinality of the set J, and e_n is the nth unit vector. int$X$ denotes the interior of the set X.

4.2 Why infeasibility handling is needed

Let the system to be controlled be described by

$$x_{t+1} = f(x_t, u_t, \eta_t), \tag{4.1}$$

for some $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^{n_\eta} \to \mathbb{R}^n$, where x_t, u_t, and η_t denote the state-, control,- and disturbance-vector at time t respectively. The presentation is based on the well known linear MPC problem (Rawlings and Muske, 1993):

$$\begin{array}{ll} \min_{\bar{\pi}_t} \phi(x_t, \bar{\pi}_t) = \sum_{j=t}^{\infty} x_{j|t}^T Q x_{j|t} + u_{j|t}^T R u_{j|t} & \\ \text{subject to:} & \\ \quad x_{t|t} = x_t & \\ \quad x_{t+N|t}^u = 0 & \\ \quad x_{j+1|t} = Ax_{j|t} + Bu_{j|t}, \; t \leq j & \\ \quad Hx_{j|t} \leq h, \qquad t < j \leq j_2 + t & \\ \quad Du_{j|t} \leq d, \qquad t \leq j & \\ \quad u_{j|t} = 0 \qquad t + N \leq j & \end{array} \tag{4.2}$$

where $Q \geq 0$, $R > 0$, $\bar{\pi}_t = (u_{t|t}, \cdots, u_{t+N-1|t})$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, and $x_{j|t} \in \mathbb{R}^n$, $u_{j|t} \in \mathbb{R}^m$, are the predicted state and control input vector at future time j, respectively, and $x_{j|t}^u \in \mathbb{R}^{n_u}$ denote the unstable modi of the predictor at future time j. The predictor is given by the third constraint in (4.2). Further, $H \in \mathbb{R}^{n_h \times n}$, $h > 0 \in \mathbb{R}^{n_h}$, $D \in \mathbb{R}^{n_d \times m}$, and $d > 0 \in \mathbb{R}^{n_d}$. Assume that (A, B) is stabilizable and $N \geq \max\{n_u, 1\}$. Due to the constraint $u_{j|t} = 0, t+N \leq j$, there exists a constraint horizon $j_2 \geq N$ such that satisfaction of $Hx_{j|t} \leq h,\ t < j \leq j_2 + t$ implies $Hx_{j|t} \leq h,\ t < j$ (Rawlings and Muske, 1993). Note that the system (4.1) and the predictor are equal if $f(x, u, \eta) = Ax + Bu$. This defines the nominal case.

The (linear) inequality constraints in (4.2) may be restated as

$$S\bar{\pi}_t \leq s_0 + S_1 x_t, \tag{4.3}$$

where S and S_1 are matrices, and s_0 is a vector. Note that the right hand side of (4.3) is parameterized by the state x_t. Thus, due to e.g. disturbances, operator interventions, modelling errors or plant failures, the state may take a value such that (4.3) has no solution (i.e. the MPC optimization problem is infeasible). If the operators are allowed use elements of h and d as on-line tuning parameters, the corresponding elements of h and d appears as parameters in (4.3) similar to x_t. In the following, for the ease of presentation, we assume that h and d are constants. However, the proposed infeasibility handler can be used for time-varying h and d as well (operator intervention).

The constraints in an MPC optimization problem can be divided into the following classes:

Non-relaxable hard constraints: Hard inequality constraints that are absolute in the sense that they cannot under any circumstances be violated. Constraints related to physical limitations belongs to this class.

Relaxable hard constraints: Hard inequality constraints related to desirables. These constraints are relaxed only in cases when the whole set of hard constraints (i.e. relaxable and non-relaxable) is inconsistent.

Soft constraints: Inequality constraints related to desirables. Violation of these constraints are allowed, but a term is included in the cost function (4.2) which penalizes constraint violations (see e.g. (Zheng and Morari, 1995), or (Scokaert and Rawlings, 1999)).

End point constraints: (I.e. $x^u_{t+N|t} = 0$.) These are equality constraints related to stability. If these constraints are violated, nominal stability is not guaranteed (Rawlings and Muske, 1993).

Note that, generally, soft constraints, as opposed to relaxable hard constraints, can be violated even if this is not necessary in order to obtain a feasible solution to the original MPC optimization problem. In (Scokaert and Rawlings, 1999), the concept of exact soft constraints is introduced. This is a strategy to relax the soft constraints only when this is necessary in order to obtain a feasible solution of the MPC optimization problem. If this strategy is used, the soft constraints can be considered as relaxable hard constrains.

In the rest of this paper, unless otherwise stated, we make the following assumptions:

- There are no soft constraints.
- The MPC optimization problem is always feasible when all relaxable hard constraints are removed. (If this assumption is violated, some extraordinary action like shutdown, or switching to manual control, is required).
- All necessary degrees of freedom are used to minimize the violation of the constraints, and all, if any, remaining degrees of freedom are used to minimize the cost function in (4.2).

4.3 How to recover from infeasibility

Given an initial state x_t such that the MPC optimization problem (4.2) is infeasible, and assume that there exists a $\bar{\pi}_t$ such that the non-relaxable hard constraints in (4.2) can be satisfied. One strategy to obtain a feasible optimization problem is simply to treat all relaxable hard constraints in an equal manner and remove all of them (Scokaert, 1994). This strategy is computationally cheap. It may, however, lead to unnecessary and unacceptable large constraint violations. A refinement of this strategy is to relax all relaxable hard constraints with the same amount, i.e. an amount sufficiently large to guarantee feasibility. This refinement also has the disadvantage that it may result in unnecessary large constraint violations. In neither of these strategies it is possible to impose that some relaxable constraints are more important to fulfill than others. All other strategies use some kind of discrimination of the constraints, i.e. they use some kind of prioritization among the constraints. Such strategies is a natural way to implement certain types of operational objectives of the type "it is more important to avoid triggering an alarm than it is to keep the engine speed within its optimal interval". We divide the infeasibility handling strategies involving prioritization into two classes:

Hard prioritization: The prioritization among the constraints is absolute, i.e. a higher prioritized constraint is "infinitely" more important to fulfill than a lower prioritized constraint. One way to recover from infeasibility is simply to remove a sufficiently large subset of the constraints, where the members of this subset are determined by the prioritization (Scokaert, 1994). However, this strategy has the drawback that it generally leads to unnecessary large constraint violations. A rigorous extension to this strategy is to compute the minimal relaxation of the constraints in this subset according to the prioritization among the constraints in this subset, (Tyler and Morari, 1999), (Scokaert, 1994), (Meadowcroft et al., 1992), (Vada, Slupphaug and Foss, 1999) and (Vada, Slupphaug and Johansen, 1999b).

Soft prioritization: The prioritization among the constraints is not absolute. When the MPC optimization problem becomes infeasible, the original MPC cost function is extended with a penalty function which penalizes the constraint violations. The individual weights on the different violations determine the relative importance of each constraint. (See e.g. (Zheng and Morari, 1995), or (Scokaert and Rawlings, 1999)).

Compared to hard prioritization, an advantage obtained by using soft prioritization is that it is straightforward to implement and it gives only a small increase in the on-line computational load. However, it is not straightforward to choose the weights so as to obtain the desired prioritization. In (Tyler and Morari, 1999) this problem is illustrated by an example, where a penalty function is designed to give a certain prioritization for a given disturbance. However, for another disturbance (of similar size), the constraint violations obtained by using this is not optimal according to this prioritization. Such design difficulties is one of the reasons why we in the following concentrate on hard prioritization. Moreover, using hard prioritization, the relation between the specification and the achieved prioritization is explicit, and thus the design difficulties experienced by using soft prioritizations are not present.

According to (Meadowcroft et al., 1992), "if a designer can capture the desired performance in a utility (single-objective) function, then automatically he/she has also assigned priority levels to the various objectives, but the reverse is not always true". In the next section, we show that for the linear case, the reverse is in fact true. The on-line computational complexity of the approach we follow is comparable to the on-line computational complexity of a similar soft prioritization approach.

4.4 Optimal weight design problem (OWDP)

In this section we formulate the problem of computing optimal constraint violations subject to hard prioritization as a single LP problem to be solved on-line at each sample. It is non-trivial to see that this is indeed possible, but this question was indeed solved in (Vada, Slupphaug and Johansen, 1999a).

The constraints in the MPC optimization problem (4.2) can be transformed into the following three constraint sets:

$$\begin{array}{rl} G^1\pi_t & = g^1(x_t),\ g^1(x_t) := g^{10} + g^{11}x_t \\ G^2\pi_t & \leq g^2(x_t),\ g^2(x_t) := g^{20} + g^{21}x_t \\ G^3\pi_t & \leq g^3(x_t),\ g^3(x_t) := g^{30} + g^{31}x_t \\ \pi_t & \geq 0 \end{array} \tag{4.4}$$

where $G^1 \in \mathbb{R}^{n_u \times m \cdot N}$, $G^2 \in \mathbb{R}^{m_2 \times m \cdot N}$, $G^3 \in \mathbb{R}^{m_3 \times m \cdot N}$, $g^{10} \in \mathbb{R}^{n_u}$, $g^{11} \in \mathbb{R}^{n_u \times n}$, $g^{20} \in \mathbb{R}^{m_2}$, $g^{11} \in \mathbb{R}^{m_2 \times n}$, $g^{30} \in \mathbb{R}^{m_3}$, $g^{31} \in \mathbb{R}^{m_3 \times n}$, and $\pi_t := \bar{\pi}_t - \pi^{\min} \in \mathbb{R}^{m \cdot N}$ is a modified vector of control inputs, where $\pi^{\min}$ is the lower limit on each control input. Such a limit will always be present in a practical MPC problem, since each element of $\bar{\pi}_t$ is related to a physical quantity. (However, if for some reason, $\pi_i^{\min}$ does not exist, just replace π_i with $u_i - v_i$ in (4.4), with $u_i^{\min} = v_i^{\min} = 0$.) In (4.4), $G^1\pi_t = g^1(x_t)$ corresponds to the stability constraint $x^u_{t+N|t} = 0$ in (4.2). Further, the inequality constraints in (4.2) are partitioned into the following two sets of constraints:

$G^2\pi_t \leq g^2(x_t)$, which is the set of all non-relaxable hard constraints, and $G^3\pi_t \leq g^3(x_t)$, which is the set of all relaxable hard constraints. The total number of inequality constraints in (4.2) is $n_d \cdot N + n_h \cdot j_2$, and thus $m_2 + m_3 = n_d \cdot N + n_h \cdot j_2$. The relation between (4.4) and (4.2) is easy established by, in (4.2), inserting the 1st, 3rd, and 6th constraint into the 2nd, 4th, and 5th constraint and by replacing $\bar{\pi}_t$ with $\pi_t + \pi^{\min}$. This is detailed in (Vada, Slupphaug and Johansen, 1999a). Further, assume that there exists a hard prioritization among the inequalities in $G^3\pi_t \leq g^3(x_t)$, and that G^3 and g^3 are constructed such that the ith row of $G^3\pi_t \leq g^3(x_t)$ have higher priority than the $(i+1)$th row. This implies that minimizing the violations of the ith row of $G^3\pi_t \leq g^3(x_t)$ is "infinitely" more important than minimizing the violations of the $(i+1)$th row.

Assume that, at a given sample, the optimization problem in (4.2) is infeasible, that is, there is no feasible solution to (4.4). Since the 3rd constraint in (4.4) is the only relaxable hard constraint, in order to transform (4.2) into a feasible optimization problem, we introduce a vector of constraint violations $z_t \in \mathbb{R}^{m_3}$ as follows

$$\begin{aligned} G^1\pi_t &= g^1(x_t) \\ G^2\pi_t &\leq g^2(x_t) \\ G^3\pi_t &\leq g^3(x_t) + z_t \\ \pi_t, z_t &\geq 0, \end{aligned} \tag{4.5}$$

Next we introduce the notion of *lexicographic minimum*: $y^o \in Y \subseteq \mathbb{R}^n$ is the lexicographic minimum of Y if it is not possible to find another $y \in Y$ and an $i \in \mathbb{I}_n^+$ such that $y_i < y_i^o$ and $y_j = y_j^o$, $j \in \mathbb{I}_{i-1}^+$. As an example $[0.10, 0.01, 10000]$ is lexicographically less than $[0.10, 0.011, 0]$, since the first element of both vectors are equal, while minimizing the second element is "infinitely" more important than minimizing the third.

Now we are ready to state a problem whose solution can be used to compute optimal constraint violations (according to the given hard prioritization) by solving only one LP problem on-line in addition to the original MPC QP problem:

Optimal weight design problem (OWDP)
Let $X \neq \emptyset$ denote the set of all x_t such that there exists (π_t, z_t) satisfying (4.5). Given an $x_t \in X$, let $Z(x_t)$ denote the set of all $z_t \geq 0$ such that there exists a π_t satisfying the inequalities (4.5). Design the weight vector $\tilde{c}$ in (4.6) such that $\forall x_t \in X$, z_t^* defined by

$$(\pi_t^*, z_t^*) := \text{argmin } \tilde{c}^T z \quad \textit{subject to (4.5)}, \tag{4.6}$$

is equal to the lexicographic minimum of $Z(x_t)$.

Note that in the OWDP, for a given x_t, z_t^*, the optimum of (4.6), represents the optimal constraint violations of the constraints in (4.2) with respect to the given hard prioritization. That is, there does not exist any $z_t \in Z(x_t)$ which violates these constraints less with respect to the given hard prioritization. Also note that since we have assumed that $d, h > 0$, (A, B) stabilizable, and $N \geq \max\{n_u, 1\}$, we have that $X \neq \emptyset$ and $0 \in \text{int}X$. In (Vada, Slupphaug and Johansen, 1999a), existence of a solution to the OWDP under these assumptions is established, and in (Vada, Slupphaug and Johansen, 1999b) it is shown how the OWDP may be solved. A consequence of this result is: At each sample, if *i*) the state x_t has a value making (4.2) infeasible, and *ii*) $x_t \in X$, that is, with the given x_t there exists a relaxation of the relaxable hard constraints such that (4.2) becomes feasible, then an optimal relaxation z_t^* can be computed by solving the LP problem in (4.6).

Note the similarity between the proposed infeasibility handler and the optimal minimal time approach (Scokaert and Rawlings, 1999), where the minimal horizon $\kappa(x_t)$ beyond which all hard state constraints can be satisfied is first computed (only the state constraints can be relaxed). Next, the constraint relaxation on the first $\kappa(x_t)$ samples on the horizon is minimized by solving an LP problem similar to (4.6). The similarity is that in both approaches the optimal constraint relaxations are computed by solving a separate problem, and next, the original cost function in the MPC problem is solved subject to the corresponding modified constraints. However, there are three important differences between this approach and the one we propose.

- In the optimal minimal time approach, there is no way to implement hard prioritization between the different rows of $Hx_{j|t} \leq h$, since $\kappa(x_t)$ is common to all rows.
- In the optimal minimal time approach, in order to compute the optimal constraint relaxations, at each sample a sequence of optimization problems is first solved in order to compute $\kappa(x_t)$. Then, in order to compute the optimal constraint relaxations on the first $\kappa(x_t)$ samples, a single LP problem (similar to (4.6)) is solved. In the approach we propose, only one LP problem is solved at each sample.
- In the optimal minimal time approach, there is an LP which minimizes a weighted ∞-norm of the constraint relaxations on the first $\kappa(x_t)$ samples, while in the present approach, a weighted l_1-norm of all constraint relaxations on the whole constraint horizon is minimized. Thus, the optimal minimal time approach can be classified

as an approach with two hard priority levels: All constraint on samples beyond $\kappa(x_t)$ are collected in the highest priority level, and all other constraints are collected in the lowest priority level. However, within the lowest priority level, there is a soft prioritization among the constraint violations.

4.5 Stability

In this section we show that by combining the proposed infeasibility handler with the MPC controller defined in Section 4.2, the region of attraction of the original MPC controller without an infeasibility handler is increased. For a certain prioritization, Theorem 4.1 below establishes nominal asymptotic stability for the receding horizon implementation of (4.2) if the constraints in (4.2) are replaced by (4.5) with $z_t = z_t^*$, where the weights $\tilde{c}$ in (4.6) is a solution to the OWDP. First, we present Lemma 4.1 which is needed in the proof of Theorem 4.1:

LEMMA 4.1
Assume that the constraints $x_{t+j|t} \in \tilde{X}, \forall j > 1$, are hard non-relaxable constraints in (4.2), where $\tilde{X} \subset \mathbb{R}^n$, $0 \in \text{int}\tilde{X}$, is an arbitrary bounded subset of X. Then, in (4.2), there exists a sufficiently large $j_2 \geq N$ such that $\forall x_t \in \tilde{X}$, $Hx_{t+j_2|t} \leq h \Rightarrow Hx_{t+j_2+i|t} \leq h$, $i = 1, 2, \dots$.

Proof: Follows from (Rawlings and Muske, 1993) and boundedness of $\tilde{X}$.
□

Next, we define a prioritization among the constraints which is used in Theorem 4.1:

Priority Assumption
Assume that a unique priority level is assigned to each relaxable row of $Hx_{j|t} \leq h$ and $Du_{j|t} \leq d$ in (4.2), such that all constraints on the horizon related to a certain relaxable row of $Hx_{j|t} \leq h$ or $Du_{j|t} \leq d$ with a given priority level have higher priority than any constraint on the horizon related to rows with a lower priority level . Let H_i (D_i) denote the ith row of H (D), and assume that $\forall j \in \mathbb{I}^+_{j_2-1}, i \in \mathbb{I}^+_{n_h}, t \geq 0$, $H_i x_{t+j+1|t} \leq h_i$ has higher priority than $H_i x_{t+j|t} \leq h_i$, and that $\forall j \in \mathbb{I}^+_{N}, i \in \mathbb{I}^+_{n_d}, t \geq 0$, $D_i u_{t+j|t} \leq h_i$ has higher priority than $D_i u_{t+j-1|t} \leq d_i$.

THEOREM 4.1
Assume that the constraints $x_{t+j|t} \in \tilde{X}$, $\forall j > 1$ are hard non-relaxable constraints in (4.2), and let j_2 be given as in Lemma 4.1. Let G^3 and g^3 in

(4.5) be constructed according to the Priority Assumption. Assume further that i) $f(x, u, \eta) = Ax + Bu$, ii) $\forall t \geq 0$, $z_t = z_t^*$, *the solution to (4.6), where* $\tilde{c}$ *is a solution to the OWDP, and* iii) $\forall t \geq 0$, $u_t = u_{t|t}^*$, *where* $u_{t|t}^*$ *is the first* m *elements of the solution of (4.2) where the constraints are replaced by (4.5). Then,* $\forall x_0 \in \tilde{X}$, $\{z_t\}_{t=0}^{\infty}$ *becomes zero within finite time, and the origin is an asymptotically stable equilibrium point with* $\tilde{X}$ *contained in the region of attraction.*

Proof: First we prove that z_t^* becomes zero in finite time: Given any $x_0 \in \tilde{X}$, let z_0' denote the constraint violations obtained by shifting the constraint violations in z_0^* one step ahead and filling up with zeros in the locations corresponding to prediction $j_2 + 1|1$ $(N|1)$ for the state (control input) constraints. Thus, since $x_1 = x_{1|0}$ (nominal case), at time $t = 1$, $z_1 = z_0'$ is feasible, and hence zero violation of the constraint at the end of the horizon of the highest prioritized constraint is feasible. Then, due to the choice of $\tilde{c}$ in (4.6), the corresponding element of z_1^* is equal to zero. Continuing this argument, due to the prioritization along the horizon, we obtain that in z_t^*, all violations of $Hx_{j|t} \leq h$ (or $Du_{j|t} \leq d$) corresponding to the highest priority level, becomes zero after at least j_2 (or N) samples. Continuing this for the row of $Hx_{j|t} \leq h$ or $Du_{j|t} \leq d$ corresponding with the next priority level, and so on, we obtain that $z_t^* = 0_{m_3}$, $t = m_3, m_3 + 1, \ldots$.

Finally we prove asymptotic stability with $\tilde{X}$ contained in the region of attraction: Let X' be the set of all x_t such that there exists a π_t satisfying (4.5) with $z_t = 0$. It follows from (Rawlings and Muske, 1993) that $\forall x \in X'$, by using the control law defined by receding-horizon implementation of the solution of (4.2), the origin is an asymptotically stable solution. Combining this with the fact that $z_t^* = 0_{m_3}$, $t = m_3, m_3 + 1, \ldots$, the result follows. □

Note that we need to assume that $\forall t \geq 0$, x_t is contained in a bounded region $\tilde{X}$. This is done to obtain a fixed j_2 which is sufficiently large to be valid for all $x_t \in \tilde{X}$. Also note that a result similar to Theorem 4.1 is stated in (Rawlings and Muske, 1993) and (Scokaert and Rawlings, 1999) for the case when only the state constraints can be relaxed, and when all rows of $Hx_{j|t}$ have equal priority. Recall that the strategy in (Scokaert and Rawlings, 1999) is based on solving a sequence of optimization problems. An important consequence of Theorem 4.1 is that by using the proposed controller, the region of attraction of the MPC controller (4.2) without infeasibility handling is at least enlarged from X' to $\tilde{X}$ (cf. the proof of Theorem 4.1 for X'). Finally note that, in the case when all state constraints and none of the input constraints are relaxable, the region of attraction of the approaches proposed in (Rawlings and Muske, 1993) and (Scokaert and Rawlings, 1999) (the optimal minimal time approach) are equal to the region of attraction obtained by using the infeasibility handler we propose.

4.6 Solving the OWDP

In (Vada, Slupphaug and Johansen, 1999b), an algorithm which solves the OWDP is presented, along with some new contributions to field of parametric programming and multi-objective linear programming (MOLP) used in the algorithm. In order to give an intuitive understanding of this algorithm, we give in the following an outline of the main ideas behind the algorithm.

The constraints in the OWDP, i.e. (4.5), can be restated as a set of equality constraints by introducing nonnegative auxiliary variables v_t and w_t

$$\begin{aligned} G^1\pi_t &= g^1(x_t) \\ G^2\pi_t + v_t &= g^2(x_t) \\ G^3\pi_t + w_t - z_t &= g^3(x_t) \\ \pi_t, v_t, w_t, z_t &\geq 0, \end{aligned} \tag{4.7}$$

and by defining $x^{LP} := (\pi_t, v_t, w_t, z_t)$, (4.6) can be transformed into an LP problem on standard form

$$\begin{aligned} &\min_{x^{LP}} c^T x^{LP} \\ &\text{subject to:} \quad \begin{cases} A^{LP}x^{LP} = b(x_t) \\ x^{LP} \geq 0 \end{cases} \end{aligned} \tag{4.8}$$

where $b(x_t) := (g^1(x_t), g^2(x_t), g^3(x_t))$, $c := (0_{Nm+m_2+m_3}, \tilde{c})$ and

$$A^{LP} := \begin{bmatrix} G^1 & 0_{m_1\times m_2} & 0_{m_1\times m_3} & 0_{m_1\times m_3} \\ G^2 & I_{m_2} & 0_{m_2\times m_3} & 0_{m_2\times m_3} \\ G^3 & 0_{m_3\times m_2} & I_{m_3} & -I_{m_3} \end{bmatrix} \in \mathbb{R}^{(m_1+m_2+m_3)\times(Nm+m_2+2m_3)}. \tag{4.9}$$

The problem (4.8) is called a *parametric* LP (Gal, 1995), since the right hand side of the equality-constraints in (4.8) is parameterized by x_t. Recall that the problem stated in the OWDP is to design c in (4.8) (or, more precisely $\tilde{c}$, since $c_i = 0$, $i \in I^+_{Nm+m_2+m_3}$) such that for each $x_t \in X$, any optimal solution to (4.8) has the property that the z_t-part of this solution is equal to the lexicographically least feasible $z_t \geq 0$.

By using theory from parametric programming, it can be shown that X can be covered by a set of polytopes, where each of the polytopes is uniquely defined as $X_{B^{LP}} := \{x_t \in X \mid \left(B^{LP}\right)^{-1} (g^1(x_t), g^2(x_t), g^3(x_t)) \geq 0\}$, where

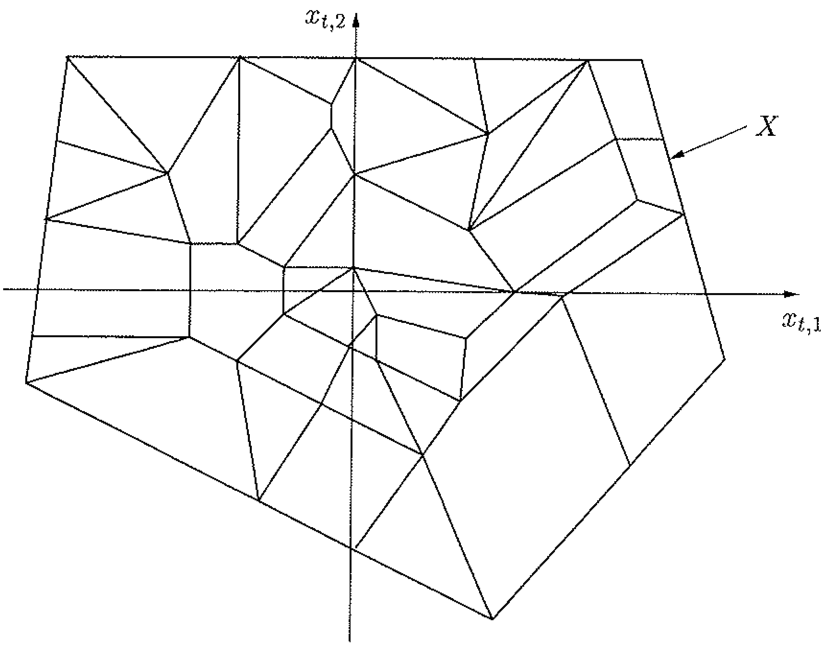

Figure 4.1: A partition of a two-dimensional state space. Each of the polytopes in the partition corresponds to a separate basis.

$B^{LP} \in \mathbb{R}^{(m_1+m_2+m_3)\times(m_1+m_2+m_3)}$ is a basis for $\mathbb{R}^{m_1+m_2+m_3}$ that consists of $m_1+m_2+m_3$ linearly independent columns of A^{LP}. Each of the polytopes $X_{B^{LP}}$ is associated with a separate basis B^{LP}, see Figure 4.1. Further, each of the bases considered has the property that if $x_t \in X_{B^{LP}}$, the non-zero elements of $z^o(x_t)$, the lexicographically minimum of $Z(x_t)$, are equal to corresponding elements of the vector $\left(B^{LP}\right)^{-1}(g^1(x_t), g^2(x_t), g^3(x_t))$. Let $\mathcal{B}$ denote the set of bases such that X is covered by the corresponding set of $X_{B^{LP}}$s. In (Vada, Slupphaug and Johansen, 1999b) it is shown that each basis in $\mathcal{B}$ defines a set of linear constraints on $\tilde{c}$ in (4.6) in order for $\tilde{c}$ to solve the OWDP. The main idea is to compute a $\tilde{c}$ which satisfies the set of constraints defined by *all* bases in $\mathcal{B}$. $\mathcal{B}$ is computed by an sequential algorithm that finds a new basis B^{LP} in $\mathcal{B}$ by moving into a new region of X defined by neighbors of already computed regions. This algorithm is continued until X is covered. Note that $\mathcal{B}$ is finite. In order to get more insight into this algorithm, see Section 4.A, where a summary of the algorithm is given. Full details can be found in (Vada, Slupphaug and Johansen, 1999b).

4.7 Practical modifications and computational issues

4.7.1 Sharing a priority level

In some MPC implementations it might not be desirable or natural to distinguish between each (scalar) constraint by assigning different priority levels to each of them. In such cases, two or more constraints can be collected into the same priority level and thus share the same element of z_t. Note that by this, the size of the OWDP is also reduced (fewer elements in $\tilde{c}$) at the cost of a possibly increased number of constraint violations. Hence, collecting several constraints into the same priority level can also be useful in cases when the computational load and memory capacity needed in order to solve the OWDP becomes prohibitively large. In order to allow for several constraints having the same priority level, consider the following modification of the original constraints, i.e. (4.5):

$$\begin{aligned} G^1\pi_t &= g^1(x_t) \\ G^2\pi_t &\leq g^2(x_t) \\ G^3\pi_t &\leq g^3(x_t) + G^z z_t^{red} \\ \pi_t, z_t^{red} &\geq 0, \end{aligned} \tag{4.10}$$

where $G^z \in \mathbb{R}^{m_3 \times m_3^{red}}$, and $z_t^{red} \in \mathbb{R}^{m_3^{red}}$, where $m_3^{red} < m_3$ is the total number of priority levels in the modified formulation. Note that in (4.5), $G^z = I_{m_3}$ and $z_t \in \mathbb{R}^{m_3}$. Let $\mathcal{I}_j$ denote the set of indices to the rows of $G^3\pi_t \leq g^3(x_t)$ having the priority level j. Constraint violation $z_{t,j}$ is shared by all these constraints if $\forall i \in \mathcal{I}_j$, row i of G^z is equal to $r_i e_j$, where $r_i > 0 \in \mathbb{R}$ is present to allow for individual scaling of $z_{t,j}$. This can be useful if the constraints within the same priority level have different units. In order to solve the OWDP with respect to this modification, in (4.8), replace A^{LP} with

$$A^{LP,red} := \begin{bmatrix} G^1 & 0_{m_1\times m_2} & 0_{m_1\times m_3} & 0_{m_1\times m_3^{red}} \\ G^2 & I_{m_2} & 0_{m_2\times m_3} & 0_{m_2\times m_3^{red}} \\ G^3 & 0_{m_3\times m_2} & I_{m_3} & -G^z \end{bmatrix}. \tag{4.11}$$

Note that by this, there are m_3^{red} weights (i.e. elements of $\tilde{c}$) to be determined in the OWDP, and the upper bound on the number of inequality constraints in the LP problem which computes $\tilde{c}$ (see Lemma 3.4) is changed from $|\mathcal{B}|(Nm-m_1+m_3)+m_3+1$ to $|\mathcal{B}^{red}|(Nm-m_1+m_3^{red})+m_3^{red}+1$, where $\mathcal{B}^{red}$ is the set of optimal bases (according to the prioritization) corresponding to the modified OWDP, and the number of variables in this LP problem is reduced from m_3+1 to $m_3^{red}+1$. Note that $|\mathcal{B}^{red}|$ will generally be different from $|\mathcal{B}|$. It is however difficult to determine whether or not $|\mathcal{B}^{red}| \leq |\mathcal{B}|$ since the number of inequalities defining $X_{B^{LP}} = \{x_t \in X \mid (B^{LP})^{-1}b(x_t) \geq 0\}$ is m_3 in both cases. Asymptotic stability of the MPC controller (4.2) combined with this infeasibility handler can be shown in the same way as in the proof of Theorem 4.1 if the prioritization along the horizon is the same as in Theorem 4.1.

Another modification of the proposed infeasibility handler which also allows for more than one constraint having the same priority level and which also reduces the size of the OWDP, is to divide the relaxable constraints into two parts: hard prioritized constraints and soft prioritized constraints. This can be done by classifying the m_3^{red} most important relaxable constraints as hard prioritized constraints and the $m_3 - m_3^{red}$ other relaxable hard constraints as soft prioritized constraints. The OWDP is then solved without the set of the soft prioritized constraints, and by this the size of the corresponding LP which computes $\tilde{c}$ is reduced equivalently as with the modification proposed above. However, note that in this case, we expect that $|\mathcal{B}^{red}|$ will be less than $|\mathcal{B}|$, since the number of inequalities defining $X_{B^{LP}} = \{x_t \in X \mid (B^{LP})^{-1}b(x_t) \geq 0\}$ is reduced from m_3 to m_3^{red}, and this is expected to give larger polytopes. The proposed infeasibility handler now becomes as follows: At each sample, compute the optimal relaxation of the m_3^{red} highest prioritized constraints by solving the reduced version of (4.6), and then compute the optimal relaxation (in some manner) of the other lower prioritized constraints by either solving a separate optimization problem minimizing some norm of the violation of these constraints, or by adding a corresponding penalty term to the cost function in (4.2). Both these optimization problems have the hard prioritized constraints and their corresponding optimal relaxations as hard non-relaxable constraints. Note that the first of these two approaches implies that more effort is used to minimize the relaxation of the constraints than in the last one. In order to establish stability for these two approaches, further investigation is required. (For stability of MPC with soft prioritization only, see e.g. (Zheng and Morari, 1995).)

Whether or not each constraint should have a separate priority level, or whether or not the lower prioritized constraints can be treated as soft constraints, is of course dependent on the application. However, it is important to note that it is not straightforward (if possible at all) to choose the weights in a soft constrained approach so as to obtain the desired hard prioritization for all possible initial states. Thus, in order to ensure that the violation of the most important relaxable constraints, such as shut-down and alarm limits (if they are relaxable), are minimized according to a given prioritization, the violation of these constraints should be computed by a hard prioritized infeasibility handler.

4.7.2 Computational issues

For some MPC applications, the (off-line) computational load and memory storage capacity required by the proposed solution strategy of the OWDP may become prohibitively large. The computational- and memory storage demands are dependent on e.g. the dynamics of the predictor, the number of states, the number of priority levels, and the hard constraints. Hence it is hard to state some general considerations about the class of prioritized infeasibility problems which can be solved by the proposed solution strategy. In this section we discuss strategies which can be used to reduce the computational demands.

The computational load and memory capacity required for computing and storing the index set to each basis is proportional to $|\mathcal{B}|$. The computational load can only be reduced by reducing the size of the original OWDP, or by designing a more efficient algorithm to compute $\mathcal{B}$. The latter issue is discussed in (Vada, Slupphaug and Johansen, 1999b), while we here focus on the first. Recall that the suggested modifications of the proposed infeasibility handler described in Section 4.7.1, which allows for several constraints having the same priority level, reduces the size of the OWDP. Another modification which also reduces the size of the OWDP is to reduce the horizon of the state constraints (i.e. j_2). Note that if j_2 is less than the minimal j_2 satisfying the condition given in Lemma 4.1, nominal stability of the controller is no longer guaranteed. Simulations indicate that the method we have used for computing j_2 give a very conservative estimate (see Section 4.8), thus much can be gained by improving the procedure for computing j_2.

Limited storage capacity can be remedied by partitioning X into several subsets X^i such that $X = \cup_{i\in\mathbb{I}^+_{N_X}} X^i$, where N_X is the number of subsets, and then compute an individual $\mathcal{B}^i$ for each subset X^i. In order to compute $\tilde{c}$ then, we propose two strategies: One strategy is to compute an individual vector of weights $\tilde{c}^i$ for each $\mathcal{B}^i$ by using the algorithm described in Section 4.6. Note that this strategy implies that the infeasibility handler must on-line detect the subset containing x_t in order to select the right $\tilde{c}^i$ to be used in (4.6). Another strategy, which results in a $\tilde{c}$ valid for all X^i, $i \in \mathbb{I}^+_{N_X}$, is to let the $\tilde{c}^i$ be defined via consecutive ratios r_k as follows: Given $\tilde{c}^i_{m_3} > 0$; $\tilde{c}^i_k := r^i_k \tilde{c}_{k+1}$, $r^i_k > 0$, $k \in \mathbb{I}^+_{m_3-1}$.

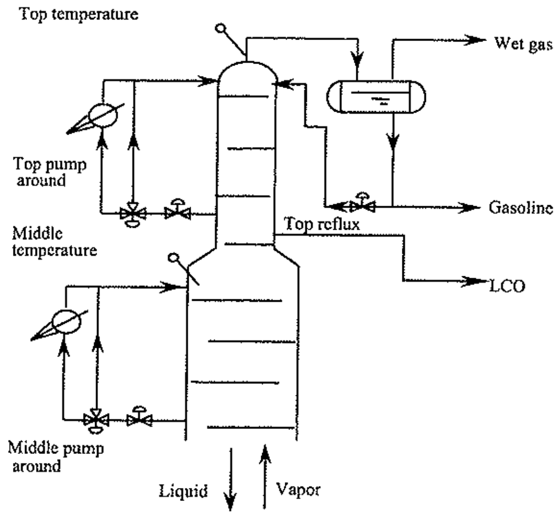

Figure 4.2: Top part of a FCCU fractionator.

In (Vada, Slupphaug and Johansen, 1999a), existence of such ratios is established, and it is shown how such ratios can be computed for a given $\mathcal{B}^i$.

4.8 Simulation example

In this section, we illustrate the use of the proposed infeasibility handler for a linear model of the top section of a fluid catalytic cracker unit (FCCU) main fractionator, see Figure 4.2, which is a critical unit for separating gasoline and LCO (diesel) from the feedstock from an upstream riser reactor.

A rigorous model of the fractionator has been developed and fitted to real plant data (Cong, Yuan and Shen, 1998), and a linear model has been derived by linearization of this model around a nominal operating point:

$$x_{t+1} = Ax_t + Bu_t, \tag{4.12}$$

Var.	Description	Unit
$(x_t)_1$	Top vapor temperature.	°C
$(x_t)_2$	Middle vapor temperature.	°C
$(x_t)_3$	Top heat exchanger outlet temperature.	°C
$(x_t)_4$	Middle heat exchanger outlet temp.	°C
$(u_t)_1$	Top pump-around tee valve position	%
$(u_t)_2$	Top reflux valve position.	%
$(u_t)_3$	Top pump-around valve position.	%
$(u_t)_4$	Middle pump-around tee valve position.	%
$(u_t)_5$	Middle pump-around valve position.	%

Table 4.1: States and control inputs.

where

$$A = 0.01 \begin{bmatrix} 90.9028 & -6.4632 & 5.7545 & 1.5616 \\ -1.2803 & 92.1012 & 1.5499 & 3.3564 \\ -0.7713 & -4.2529 & 85.3340 & 3.4314 \\ 0.2413 & -0.7122 & -0.1415 & 85.6537 \end{bmatrix},$$

$$B = 0.01 \begin{bmatrix} -3.6674 & -12.4077 & -0.0648 & -5.7626 & 0.1049 \\ -2.0582 & -0.5071 & 0.1088 & -7.2823 & 0.2324 \\ -1.3759 & -1.6691 & 1.5177 & -1.8955 & -0.1224 \\ 0.1182 & -0.0777 & -0.0002 & -0.3235 & 1.4456 \end{bmatrix},$$

and $x_t := x_t^{abs} - x^{nom}$, and $u_t := u_t^{abs} - u^{nom}$, where $x^{nom} := (x_1^{nom}, \dots, x_4^{nom})$ and $u^{nom} := (u_1^{nom}, \dots, u_5^{nom})$. (x^{nom}, u^{nom}) is the nominal operating point. The sampling time is 30 s. The legend for the states and control inputs is given in Table 4.1, and the nominal operating point in addition to the absolute upper and lower bounds are given in Table 4.2. Non-relaxable hard constraints are defined as $x^{abs,lb} \leq x_t + x^{nom} \leq x^{abs,ub}$, and $u^{abs,lb} \leq u_t + u^{nom} \leq u^{abs,ub}$. Further, the relaxable hard constraints and their corresponding priority levels are given in Table 4.3. The prioritizations are based on assumptions such as: gasoline is assumed to be more valuable than LCO (this assumption determines the prioritization between priority level 1 and 2, which are related to product quality, and between priority level 3 and 4, which are related to minimizing the content of a valuable product in a less valuable product), and high production rate has higher priority than minimizing the energy use (this assumption determines the prioritization between priority level 5 and the other relaxable input constraints). Note that since the constraint horizon in (4.2) is j_2, there are j_2 constraints corresponding to each of the above defined state constraints (both relaxable and non-relaxable), and that due to the move

x_1^{nom}	107.0 °C	$x_1^{abs,lb}$	106	$x_1^{abs,ub}$	108
x_2^{nom}	219.0 °C	$x_2^{abs,lb}$	218	$x_2^{abs,ub}$	221
x_3^{nom}	87.0 °C	$x_3^{abs,lb}$	83	$x_3^{abs,ub}$	91
x_4^{nom}	199.0 °C	$x_4^{abs,lb}$	195	$x_4^{abs,ub}$	203
u_1^{nom}	62.3 %	$u_1^{abs,lb}$	0	$u_1^{abs,ub}$	100
u_2^{nom}	0 %	$u_2^{abs,lb}$	0	$u_2^{abs,ub}$	100
u_3^{nom}	50 %	$u_3^{abs,lb}$	40	$u_3^{abs,ub}$	80
u_4^{nom}	67.9 %	$u_4^{abs,lb}$	0	$u_4^{abs,ub}$	100
u_5^{nom}	50 %	$u_5^{abs,lb}$	40	$u_5^{abs,ub}$	80

Table 4.2: Nominal operating point, lower, and upper bounds.

Pri. level	Constraint	Pri. level	Constraint
1	$(x_t^{abs})_1 \leq 107.5$	6	$(u_t^{abs})_3 \leq 55$
2	$(x_t^{abs})_2 \leq 219.5$	7	$(u_t^{abs})_5 \leq 55$
3	$(x_t^{abs})_1 \geq 106.5$	8	$(u_t^{abs})_1 \leq 67$
4	$(x_t^{abs})_2 \geq 218.5$	9	$(u_t^{abs})_4 \leq 75$
5	$(u_t^{abs})_2 \leq 5.0$		

Table 4.3: Relaxable hard constraints.

i	$\tilde{c}_i$	Corresponds to:	i	$\tilde{c}_i$	Corresponds to:
79	25.70	$(x^{abs}_{t+2\|t})_2 \leq 219.5$	120	62.91	$(x^{abs}_{t+1\|t})_1 \geq 106.5$
80	113.9	$(x^{abs}_{t+1\|t})_2 \leq 219.5$	165	1.070	$(u^{abs}_{t\|t})_2 \leq 5$
119	14.86	$(x^{abs}_{t+2\|t})_1 \geq 106.5$	180	1.283	$(u^{abs}_{t\|t})_1 \leq 67$

Table 4.4: Weights solving the OWDP with $j2 = 40$. *All other* $\tilde{c}_i$ *are equal to* 1.0.

horizon, there are N constraints corresponding to each of the above defined control input constraints (both relaxable and non-relaxable). Hence, there are several constraints related to a given priority level. The prioritization implies that minimizing the violation of any of the constraints related to priority level i has higher priority than minimizing any of the constraints related to priority level $i+1$. Assume that within a given priority level, minimizing the constraint violation at prediction $k+1$ has higher priority than at prediction k. That is, we assume the same prioritization as in the Priority Assumption (defined in Section 4.5). We have chosen $N = 5$, and by using a slight modification of (Gilbert and Tan, 1991, Algorithm 3.2) to calculate j_2, assuming that the non-relaxable hard state constraints are always satisfied, we get $j_2 = 40$. Thus, for the given example, there are $m_3 = 185$ distinct priority levels, and in the OWDP, the dimension of $\tilde{c}$ is thus 185.

In order to solve the OWDP we used Algorithm 4.4 in (Vada, Slupphaug and Johansen, 1999b), which is briefly described in Section 4.6. In the algorithm, the parameter determining the lower bound on the weights is set equal to 1.0. The number of bases in the resulting set $\mathcal{B}$ is 167, and the elements of the resulting $\tilde{c}$ which are greater than 1.0 are shown in Table 4.4. The algorithm is implemented in MATLAB with NAG Foundation Toolbox, and the computation time was about 76 minutes on a Pentium 450MHz PC with 256MB RAM. Note, however, that the computation of $\tilde{c}$ is done off-line. The on-line computational effort associated with the infeasibility handler (the LP problem in (4.6)) is typically smaller than the QP (4.2).

Note that only six of the weights are greater than their minimum value. At a first glance, since the weights related to the 78 highest prioritized constraints are all equal to the predetermined minimum value, one might think that it is remarkable that these weights solve the OWDP. However, note that all weights in Table 4.4 are related to the first or second samples on the horizon for a given priority level. Thus, for the given process, minimizing the constraint violations at the beginning of the horizon implies that the constraint violations at the end of the horizon are minimized.

i	$\tilde{c}_i$	Corresponds to:	i	$\tilde{c}_i$	Corresponds to:
2	113.9	$(x_t^{abs})_2 \leq 219.5$	5	1.070	$(u_t^{abs})_2 \leq 5.0$
3	62.91	$(x_t^{abs})_1 \geq 106.5$	8	1.282	$(u_t^{abs})_1 \leq 67$

Table 4.5: Weights solving the modified OWDP with 9 priority levels and $j_2 = 40$. All other $\tilde{c}_i$ are equal to 1.0.

(Recall that within a given priority level, the constraints corresponding to the first samples of the horizon have lower priority than the samples at the end of the horizon.) Note that this might not be the case for a different process. Further note that it is by far not intuitive to determine how large the weights should to be in order to guarantee the fulfillment of the hard prioritization. The largest weight produced by this algorithm is only two orders of magnitude larger than the smallest weights. This is in strong contrast to a heuristic approach that might rely on using a sufficiently large weight ratio between each priority level. The latter approach could lead to a numerically ill-conditioned LP problem.

The simulation result obtain by combining the proposed infeasibility handler with the closed-loop implementation of (4.2) when a state disturbance of $[-1, 2, -4, 4]^T$ enters the system at $t = 0$ is shown in Figure 4.3. In (4.2), $Q = 100I_n$ and $R = I_m$. Observe that all relaxable constraints are satisfied for all $t \geq 2$. At $t = 0$, there are 4 relaxable constraints which are violated. Two of them corresponds to the first sample of the constraints with priority level 2 and 3, and the other two corresponds to the first two samples of the constraint with priority level 9. At $t = 1$, the only constraint violation corresponds to the first sample of the constraint with priority level 9.

Table 4.5 shows the weights when, within each of the priority levels given in Table 4.3, all constraints along the horizon have the same priority. In this case, there are only 9 priority levels, i.e. $m_3^{red} = 9$. The number of bases in the resulting $\mathcal{B}^{red}$ is 243, i.e. $|\mathcal{B}^{red}|$ is almost 50% larger than obtained by using the same prioritization as in the Priority Assumption. Note that for each priority level, the weights obtained by solving this modified OWDP are equal to the largest weights over the horizon in Table 4.4. Hence, reducing the number of priority levels does not imply reduced off-line computational load. The simulation results obtained with the same disturbance as above is equal to the one obtained by using the prioritization defined by the Priority Assumption, see Figure 4.3.

Table 4.6 shows the weights when $j_2 = 5$. With this choice of j_2, the corresponding m_3 becomes 45. The number of bases in the resulting $\mathcal{B}$

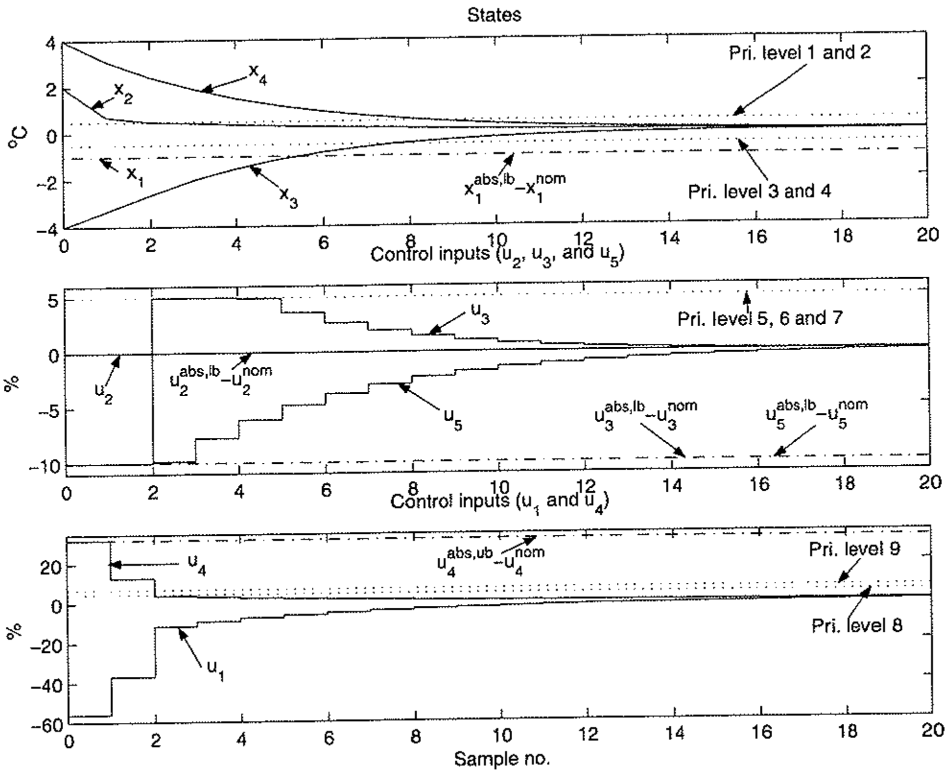

Figure 4.3: Simulation results using the proposed infeasibility handler combined with linear MPC. All values are deviations from the nominal operating point. Solid lines: states and control inputs, dash-dotted lines: non-relaxable constraints, dotted lines: relaxable constraints. Note that in the upper part, only the hard constraint on x_1 are shown, since the other hard constraints are active only at $t = 0$.

is 45, and the computation time was less than one minute. Also in this case, the simulation results obtained are equal to the one Figure 4.3, where $j_2 = 40$. This shows that for the given disturbance, the original choice of j_2 is rather conservative, and by reducing j_2 to a more realistic value, the computational load is greatly reduced. However, the first example, with $j_2 = 40$, illustrates that the the proposed solution approach can be applied on quite big problems as well.

i	$\tilde{c}_i$	Corresponds to:	i	$\tilde{c}_i$	Corresponds to:
9	25.70	$(x^{abs}_{t+2\|t})_2 \leq 219.5$	15	62.91	$(x^{abs}_{t+1\|t})_1 \geq 106.5$
10	113.9	$(x^{abs}_{t+1\|t})_2 \leq 219.5$	25	1.070	$(u^{abs}_{t\|t})_2 \leq 5$
14	14.86	$(x^{abs}_{t+2\|t})_1 \geq 106.5$	40	1.283	$(u^{abs}_{t\|t})_1 \leq 67$

Table 4.6: Weights solving the OWDP with $j_2 = 5$. All other $\tilde{c}_i$ are equal to 1.0.

4.9 Discussion/Conclusions

MPC relies on the successful solution of an optimization problem at each sample. Due to e.g. unknown disturbances, modelling errors, equipment failures, or operator interventions, this optimization problem may become infeasible. In such cases, in order to ensure that the control input can be computed, some action should be taken in order to relax the constraints of this optimization problem such that the relaxed optimization problem becomes feasible. In the present paper we consider how to optimally handle infeasibility problems caused by disturbances and operator interventions. The proposed approach can be applied in the presence of modelling errors, however, optimality is guaranteed only in the nominal case.

It is normally the case that some constraints are more important to fulfill than others. In such cases, this information defines a restriction on how the constraints should be relaxed in order to recover from infeasibility. We assume that the difference in importance can be described by the use of priority levels and we focus on how to relax the constraints when a constraint with a certain priority level is infinitely more important to satisfy than a constraint with a lower priority level. Furthermore, we assume that if a certain constraint must be violated, it is desirable to minimize the violation of this constraint.

In (Vada, Slupphaug and Johansen, 1999a), we have shown existence of a single LP problem which computes the optimal constraint violation subject to a given hard prioritization. In order to use this strategy, the cost function in this LP problem has to satisfy certain properties, and in the present paper we have given a summary of an algorithm, whose details are stated in (Vada, Slupphaug and Johansen, 1999b), which can be used to compute a cost function satisfying these properties. Further, this algorithm is applied in simulations of a realistic MPC problem, and for this problem, the computational load of the algorithm is not prohibitively large. The elements of the resulting cost function computed by the algorithm are nonintuitive, implying that designing such a cost function by trial and error might be time consuming.

There exist alternative approaches which handle infeasible MPC optimization problems with a hard prioritization among the constraints. However, to the best of the authors knowledge, in these approaches a sequence of optimization problems (LP or QP) needs to be solved at each sample. Clearly, a sequential approach is significantly more time consuming than the proposed approach where only a single LP problem needs to be solved at each sample.

In some MPC implementations it might not be desirable or natural to distinguish between each (scalar) constraint by giving them different priority levels. Hence, we propose two modifications of the infeasibility handler which assigns the same priority level to several constraints. One approach is based on hard prioritizations only, and one approach combines hard prioritization and soft prioritization. This leads to an LP with less variables, which is desirable in large-scale practical applications.

Traditionally, when designing constraints which are desirables (not related to physical limitations), one needs to consider whether or not such constraints may cause the controller to run into feasibility problems. By using the proposed approach for infeasibility handling, such considerations become less important. Actually, one might design relaxable hard constraints which one knows can be satisfied in only small regions of the state space.

The paper also proves that the proposed strategy guarantees nominal asymptotic stability if avoiding constraint violations at the end of the horizon has the highest priority. This result implies that the region of attraction of the controller without infeasibility handling is at least enlarged by using the proposed infeasibility handler.

4.A A detailed summary of the algorithm solving the OWDP

In Section 4.6, we briefly summarized the algorithm presented in (Vada, Slupphaug and Johansen, 1999b) which solves the OWDP. In the following, in order to get some more insight into this algorithm, we give a more detailed summary of this algorithm.

For a given x_t, let $\hat{Z}(x_t)$ denote the set of all $z_t \geq 0$ such that there exists a π_t, v_t, w_t satisfying (4.7). Note that $\hat{Z}(x_t) = Z(x_t)$ ($Z(x_t)$ is defined in the OWDP) and thus the lexicographic minimum of these two sets are equal. Let $z_t^o(x_t)$ denote the lexicographic minimum of $Z(x_t)$.

As may be inferred from Section 4.6, the algorithm presented in (Vada, Slupphaug and Johansen, 1999b) is based on the use of basic solutions. For completeness, we state the following definitions, which will be used in the description. A square matrix B^{LP} consisting of m^{LP} linearly independent columns of $A^{LP} \in \mathbb{R}^{m^{LP} \times n^{LP}}$ is called a *basis* for $\mathbb{R}^{m^{LP}}$. If all $n^{LP} - m^{LP}$ components of x^{LP} not associated with columns of B^{LP} are set equal to zero, the solution to $A^{LP}x^{LP} = b$ is called a *basic solution*. If, in addition, $x^{LP} \geq 0$, it is called a *basic feasible solution* and a corresponding basis is called a *feasible basis*. The components of x^{LP} associated with columns of B^{LP} are called *basic variables*. A basis whose corresponding basic solution is optimal to (4.8) is called an *optimal basis*. Let D^{LP} be defined as the columns of A^{LP} not in B^{LP}. Further, let $x_B^{LP}(x_t) \in \mathbb{R}^{m_1+m_2+m_3}$ and $x_D^{LP}(x_t) \in \mathbb{R}^{Nm-m_1+m_3}$ denote the basic and nonbasic variables of $x^{LP}(x_t)$, respectively, and let $c_B \in \mathbb{R}^{m_1+m_2+m_3}$ and $c_D \in \mathbb{R}^{Nm-m_1+m_3}$ denote the elements of c corresponding to $x_B^{LP}(x_t)$ and $x_D^{LP}(x_t)$, respectively. Further, the *relative cost vector* r_D is defined as $r_D^T := c_D^T - c_B^T(B^{LP})^{-1}D^{LP}$.

From the theory of LP it is known that nonnegativity of r_D in addition to feasibility of $x^{LP}(x_t)$ is equivalent to optimality of $x^{LP}(x_t)$ (see e.g. (Luenberger, 1989)). Note that r_D is independent on x_t. Thus, given a c and a B^{LP} such that the corresponding $r_D \geq 0$, the basic solution corresponding to B^{LP} is optimal to (4.8) $\forall x_t \in X_{B^{LP}} := \{x_t \in X \,|\, (B^{LP})^{-1}b(x_t) \geq 0\}$ (note that $X_{B^{LP}}$ is the region where B^{LP} is a feasible basis). From Lemma 3.1 it follows that $\forall x_t \in X$, there exists a basis B^{LP} such that the corresponding basic solution has the property that the $z_t(x_t)$-part is equal to $z_t^o(x_t)$.

Let $\mathcal{B}$ be a set of bases to (4.9) such that $\forall x_t \in X$, $\exists B^{LP} \in \mathcal{B}$ such that the corresponding basic solution $x^{LP}(x_t)$ is feasible and has the property that the $z_t(x_t)$-part is equal to $z_t^o(x_t)$. $\mathcal{B}$ is finite since there is a finite number of bases. Note that this implies that $X \subseteq \cup_{B^{LP} \in \mathcal{B}} X_{B^{LP}}$ (see Figure 4.1). Moreover, given a c such that $\forall B^{LP} \in \mathcal{B}$, $r_D \geq 0$. Then, $\forall x_t \in X$, there exists an optimal basic solution to (4.8) such that the $z_t(x_t)$-part of this solution is equal to $z_t^o(x_t)$. However, since an LP problem may have more than one optimal solution, in order to solve the OWDP, c must be selected such the that $z_t(x_t)$-part of *all* optimal solutions to (4.8) are equal to $z_t^o(x_t)$. In Lemma 3.4 it is shown how such a c can be computed by solving one LP problem with $m_3 + 1$ variables (the elements of $\tilde{c}$ in addition to an auxiliary variable) and at most $|\mathcal{B}|(Nm - m_1 + m_3)$ inequality constraints related to nonnegativity of r_D in addition to $m_3 + 1$ constraints related to upper and lower bounds on $\tilde{c}$. In order to avoid numerical problems when solving the LP problem (4.6), the ratio between the largest and smallest weight in $\tilde{c}$ is minimized by this algorithm.

Now it remains to show how a candidate to the set of bases $\mathcal{B}$ can be computed. The most obvious way is perhaps to compute the set of all bases to (4.9) and then check each of these bases whether or not the corresponding $X_{B^{LP}}$ is nonempty and then whether or not the corresponding basic solution has the property that the $z_t(x_t)$-part is equal to $z_t^o(x_t)$. However, the number of possible bases to (4.9) is $\binom{Nm+m_2+2m_3}{m_1+m_2+m_3}$, which is a huge number except for some very simple problems. Further, note that for a given $x_t \in X$, there might be more than one feasible basis which have the property that the $z_t(x_t)$-part is equal to $z_t^o(x_t)$. Thus, in order to reduce the computational load needed in order to solve the OWDP, it suffices to compute a subset of $\mathcal{B}$ such that X is covered by the union of the corresponding regions $X_{B^{LP}}$ (note that there may exist several such subsets).

In the case when c in (4.8) is given, Gal (1995) has developed an algorithm which, given an initial $x_t^0 \in X$ and a corresponding optimal basis B_0^{LP}, computes a set of other optimal bases $\mathcal{B}^0$ such that: *i)* $\forall B^{LP} \in \mathcal{B}^0$, $X_{B_0^{LP}}$ and $X_{B^{LP}}$ are non-overlapping, i.e. $X_{B^{LP}}$ and $X_{B_0^{LP}}$ have no common interior points, and *ii)* $\forall B \in \mathcal{B}^0$, $X_{B^{LP}}$ and $X_{B_0^{LP}}$ have one common facet, see Figure 4.4. In the figure, each polytope is marked by an integer, and the polytope marked by k corresponds to the region $X_{B_k^{LP}}$ where B_k^{LP} is an optimal basis. Note that $X_{B_5^{LP}}$ and $X_{B_6^{LP}}$ are overlapping, and thus, in this example, there are two candidates to $\mathcal{B}^0$: $\{B_1^{LP}, B_2^{LP}, B_3^{LP}, B_4^{LP}, B_5^{LP}\}$ and $\{B_1^{LP}, B_2^{LP}, B_3^{LP}, B_4^{LP}, B_6^{LP}\}$. Each of the bases in $\mathcal{B}^0$ is called a neighbor to B_0^{LP}. The neighbors are computed by using the same ideas as in the dual simplex method (see e.g. (Luenberger, 1989)). The same principle is used in Section 3.4.2 in order to develop an algorithm which computes a set of bases $\tilde{\mathcal{B}} \subseteq \mathcal{B}$ such that $\forall x_t \in X$, $\exists B^{LP} \in \tilde{\mathcal{B}}$ such that the corresponding basic solution has the property that the $z_t(x_t)$-part is equal to $z_t^o(x_t)$, and such that the corresponding $X_{B^{LP}}$s are non-overlapping. Note that in (Gal, 1995), the weights in the cost function are given, as opposed to in the OWDP, where the weights are the unknown parameters to be computed. The algorithm is sequential and adds one polyhedral region $X_{B^{LP}}$ with a corresponding basis B^{LP} at a time until the region X is covered.

Consider the following parametric preemptive multi-objective linear program defined by

$$\min_{x^{LP}} Wx^{LP}$$
$$\text{subject to } x^{LP} \in \{x^{LP} \in \mathbb{R}^{Nm+m_2+2m_3} \mid A^{LP}x^{LP} = b(x_t),\ x^{LP} \geq 0\} \tag{4.13}$$

where $W := [0_{m_3 \times (Nm+m_2+m_3)}, I_{m_3}]$, and where there exists a lexicographic ordering among the objectives (i.e. elements of Wx^{LP}), that is, $\sum_{j=1}^{Nm+m_2+2m_3} W_{i,j}x_j^{LP}$ $(= z_{t,i})$ has higher priority than

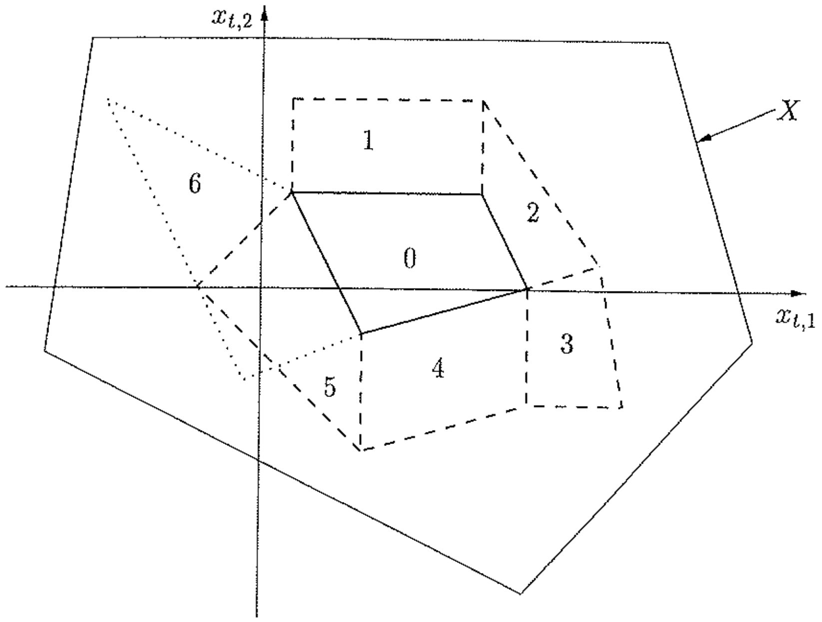

Figure 4.4: An illustration of the region where a basis is feasible.

$\sum_{j=1}^{Nm+m_2+2m_3} W_{i+1,j} x_j^{LP} (= z_{t,i+1})$. An optimal solution to (4.13) is obtained by first minimizing z_1, then minimizing z_2 while holding z_1 fixed at its minimum, and so on. Note that, given a set of weights c (or $\tilde{c}$) solving the OWDP and a $x_t \in X$, then any optimum of (4.8) is also an optimum of (4.13). Also note that each of the m_3 objectives in (4.13) have a corresponding relative cost vector r_D. Let R_D be defined as the matrix obtained by stacking the r_Ds above each other in the same order as the corresponding objectives appear in (4.13). In Lemma 3.5 it is shown that, given a basis such that the corresponding basic solution is optimal to (4.13), i.e. such that the $z_t(x_t)$-part of the corresponding basic solution is equal to $z_t^o(x_t)$, then each column of the corresponding R_D have the following property: the first non-zero element in each column is positive. In Theorem 3.3 this fact is used to extend the definition of a neighbor used in (Gal, 1995) to include preemptive MOLPs as well, along with an algorithm for computing a set of bases such that the corresponding X_{BLP}s are non-overlapping. Finally, this algorithm can be used to compute a set of bases $\tilde{\mathcal{B}}$ as follows: Compute the set of all neighbors to the bases already detected as neighbors as long as there are unexplored neighbors left.

Acknowledgment

The financial support of the Research Council of Norway, project number 107620/420, is gratefully acknowledged. Dr. Songbo Cong is also gratefully acknowledged for providing us with the linear model of the top section of the fluid catalytic cracker unit he has developed, and for giving us insight into this process.

Chapter 5

Conclusions and further work

5.1 Conclusions

In MPC, it is normally the case that some constraints are more important to fulfill than others. In such cases, this information defines a restriction on how the constraints should be relaxed in order to recover from infeasibility. It was assumed that the difference in importance can be described by the use of priority levels and focus was placed on how to relax the constraints when a constraint with a certain priority level is infinitely more important to satisfy than a constraint with a lower priority level. Furthermore, it was assumed that if a certain constraint must be violated, it is desirable to minimize the violation of this constraint.

The most obvious way to compute the optimal constraint relaxations subject to a given hard prioritization is to solve a sequence of optimization problems, where a single constraint (or a set of constraints) is relaxed at each step until feasibility is achieved. Sequential approaches are considered in Chapter 2. However, sequential approaches may be very time consuming on-line, and in many applications, only a short period of time is available to compute the control input. This is the main motivation behind the problem formulation in Chapter 3, where a rigorous method is provided for designing efficient optimal infeasibility handlers for optimization problems where there exists a hard prioritization among the constraints. The case where all constraints are linear is considered and the existence of a single LP problem is revealed which can replace the sequential solution approach, and consequently, the on-line computational load is significantly reduced. The main issue is to compute the weights in this LP problem such that its optimal solution is equal to the optimal constraint violation according to the given prioritization, and such that the same weights are valid for a user-defined set of states and constraint limits. An (off-line) algorithm for computing these weights is proposed, and thus, the only on-line computational load required by this infeasibility handler is the solution of a single LP problem. To the best of my knowledge, this is the only infeasibility handler which optimally handles hard prioritized constraints without the use of a sequential solution approach. Thus, a conclusion is that optimal infeasibility handling under prioritized constraints is efficiently solved provided the off-line computational load is not prohibitively large.

The practical viability of the proposed infeasibility handler is established in Chapter 4, where it is applied to a realistic MPC problem. For this problem, the off-line computational load is not prohibitively large. Further, the resulting weights computed by the proposed algorithm are non-trivial, hence they cannot be chosen in a direct manner. The largest weight produced by the algorithm is only two orders of magnitude larger than the smallest weights, which is remarkably small given the fact that these weights represents the relative hard prioritization between 185 constraints. An alternative to using the proposed algorithm is to compute the weights by trial and error. However, such a strategy is time consuming, and it cannot guarantee that the hard prioritization is satisfied. Typically, such a heuristic approach relies on using a sufficiently large weight ratio between each priority level, with the drawback that the resulting weights could lead to a numerically ill-conditioned LP problem.

The MPC example in Chapter 4 also illustrates that the off-line computational load required by the algorithm that is proposed to compute the weights can be significantly reduced by reducing the length of the horizon for the state constraints. Further, this example shows that assigning several constraints to the same priority level does not necessarily imply reduced off-line computational load.

5.2 Further work

There is probably a great potential for improving certain parts of the off-line algorithm proposed in Chapter 3 in order to reduce the computational load of this algorithm. Further, in order to reveal possible limitations of this algorithm, and further evaluate the proposed enhancements in Chapter 4, the proposed infeasibility handler should be applied to various MPC problems.

Compared to soft prioritization, the advantage of using hard prioritization is that it provides an explicit way of how to compute the constraint relaxations in order to obtain a feasible MPC optimization problem. It might be that it is relevant to ask the question of the desired "hardness" of the given prioritization. However, such issues are not considered in this thesis.

The proposed approach can be applied in the presence of modelling errors, however, optimality is guaranteed only in the nominal case. A topic for future work is to investigate different aspects related to robustness, in particular methods for designing weights that take into account model uncertainty.

The present work does not consider infeasibility handlers for nonlinear MPC. It should be investigated whether or not parts of the theory presented this thesis can be extended in order to design efficient infeasibility handlers for nonlinear MPC as well.

Bibliography

Allgöwer, F., Badgwell, T. A., Qin, J. S., Rawlings, J. B. and Wright, S. J.: 1999, Nonlinear Predictive Control and Moving Horizon Estimation–An Introductory Overview, *in* P. M. Frank (ed.), *Advances in Control: Highlights of ECC'99*, Springer.

Alvarez, T. and de Prada, C.: 1996, Feasibility in Constrained Predictive Control, *Proceedings of CESA'96 IMACS Multiconference*, Vol. 1, IEEE.

Alvarez, T. and de Prada, C.: 1997, Handling Infeasibilities in Predictive Control, *Computers chem. Engng.* **21**, 577–582.

Bemporad, A. and Morari, M.: 1999, Control of systems integrating logic, dynamics, and constraints, *Automatica* **35**(3), 407–427.

Cong, S. B., Yuan, P. and Shen, F.: 1998, An integrated non-equilibrium dynamic model of petroleum distillation column. Unpublished manuscript.

Gal, T.: 1995, *Postoptimal Analyses, Parametric Programming, and Related Topics*, 2 edn, Walter de Gruyter.

Garcia, C. E. and Morshedi, A. M.: 1986, Quadratic Programming Solution of Dynamic Matrix Control (QDMC), *Chemical Engineering Communications* **46**, 73–87.

Garcia, C. E. and Prett, D. E.: 1986, Advances in Industrial Model-Predictive Control, *in* M. Morari and T. J. McAvoy (eds), *Chemical Process Control - CPC III*, CACHE, pp. 249–294.

Gilbert, E. G. and Tan, K. T.: 1991, Linear Systems with State and Control Constraints: The Theory and Application of Maximal Output Admissible Sets, *IEEE Transactions on Automatic Control* **36**(19), 1008–1020.

Korhonen, P. and Halme, M.: 1996, Using Lexicographic Parametric Programming for Searching a Non-dominated Set in Multiple-Objective Linear Programming, *Journal of Multi-criteria Decision and Analysis* **5**, 291–300.

Lee, J. H.: 1996, Recent Advances in Model Predictive Control and other Related Areas, *CPC-V. Tahoe City, CA, USA*.

Luenberger, D. G.: 1989, *Linear and Nonlinear Programming*, Addison Wesley Publishing Company, Inc.

Meadowcroft, T. A., Stephanopoulos, G. and Brosilow, C.: 1992, The Modular Multivariable Controller. 1. Steady State Properties, *AIChE Journal* **38**(8), 1254–78.

Murty, K. G.: 1983, *Linear Programming*, John Wiley & Sons, Inc.

Muske, K. R.: 1995, *Linear Model Predictive Control of Chemical Processes*, PhD thesis, University of Texas at Austin.

Muske, K. R. and Rawlings, J. B.: 1993, Model Predictive Control with Linear Models, *AIChE Journal* **39**(2), 262–87.

Oliveira, N. M. C. and Biegler, L. T.: 1994, Constraint Handling and Stability Properties of Model-Predictive Control, *AIChE Journal* **40**(7), 1138–1115.

Qin, S. J. and Badgwell, T. A.: 1997, An Overview of Industrial Model Predictive Control Technology, *in* J. C. Kantor, C. E. Garcia and B. Carnahan (eds), *Fifth International Conference on Chemical Process Control*, AIChE Symposium Series 316, pp. 232–256.

Rawlings, J. B., Meadows, E. S. and Muske, K. R.: 1994, Nonlinear Model Predictive Control: A Tutorial and Survey, *Preprints IFAC Symposium ADCHEM*, Kyoto, Japan.

Rawlings, J. B. and Muske, K. R.: 1993, The Stability of Constrained Receding Horizon Control, *IEEE Transactions on Automatic Control* **38**(10), 1512–16.

Richalet, J.: 1993, Industrial Applications of Model Based Predictive Control, *Automatica* **29**(5), 1251–74.

Scokaert, P. O. M.: 1994, *Constrained predictive control*, PhD thesis, University of Oxford, UK.

Scokaert, P. O. M. and Rawlings, J. B.: 1998, Constrained linear quadratic regulation, *IEEE Trans. Auto. Cont* **43**(8), 1163–1169.

Scokaert, P. O. M. and Rawlings, J. B.: 1999, Feasibility issues in model predictive control, *AIChE Journal* **45**(8), 1649–59.

Sherali, H. D.: 1983, Preemptive and Nonpreemptive Multi-Objective Programming: Relationships and Counterexamples, *Journal of Optimization Theory and Applications* **39**(2), 172–186.

Slupphaug, O., Vada, J. and Foss, B. A.: 1997, MPC in Systems with Continuous and Discrete Control Inputs, *Proceedings American Control Conference, Albuquerque, NM, USA*.

Steuer, R. E.: 1986, *Multiple criteria optimization*, 2 edn, John Wiley & Sons, Inc.

Tyler, M. L. and Morari, M.: 1997, Propositional Logic in Control and Monitoring Problems, *Proceedings of European Control Conference '97*, Bruxelles, Belgium, pp. 623–628.

Tyler, M. L. and Morari, M.: 1999, Propositional logic in control and monitoring problems, *Automatica* **35**, 565–582.

Vada, J., Slupphaug, O. and Foss, B. A.: 1999, Infeasibility Handling in Linear MPC subject to Prioritized Constraints, *Preprints of 14th World Congress of IFAC, Beijing, China*, Vol. N, pp. 163–168.

Vada, J., Slupphaug, O. and Johansen, T. A.: 1999a, Efficient Infeasibility Handling in Linear MPC subject to Prioritized Constraints, *Proceedings of European Control Conference '99*, European Union Control Association, Karlsruhe, Germany. Paper ID: F395.

Vada, J., Slupphaug, O. and Johansen, T. A.: 1999b, Efficient Optimal Prioritized Infeasibility Handling in Model Predictive Control - a Parametric Preemptive Multi-Objective Linear Programming Approach. Submitted to Journal of Optimization Theory and Applications.

Vada, J., Slupphaug, O., Johansen, T. A. and Foss, B. A.: 1999, Stabilizing Linear MPC with Efficient Prioritized Infeasibility Handling. Accepted for presentation at ADCHEM 2000.

Vada, J., Slupphaug, O., Johansen, T. A. and Foss, B. A.: 2000, Linear MPC with Prioritized Optimized Infeasibility Handling - Application, Computational Issues and Stability. Submitted to Automatica.

Yu, P. L. and Zeleny, M.: 1975, The Set of All Nondominated Solutions in Linear Cases and a Multicriteria Simplex Method, *Journal of Mathematical Analysis and Applications* **49**(2), 430–468.

Zheng, A. and Morari, M.: 1995, Stability of Model Predictive Control with Mixed Constraints, *IEEE Transactions on Automatic Control* **40**(10), 1818–23.