A NOVEL APPROACH FOR THE COORDINATION OF BLOCK MPC

A NOVEL APPROACH FOR THE COORDINATION OF BLOCK DECENTRALIZED MODEL PREDICTIVE CONTROL

by

ALBERTO JORGE OLVERA SALAZAR, B.Eng., M.Sc.

A Thesis Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Applied Science

McMaster University

MASTER OF APPLIED SCIENCE (2009) McMaster University

(Chemical Engineering) Hamilton, Ontario, Canada

TITLE: A Novel Approach for the Coordination of Block Decentralized Model Predictive Control

AUTHOR: Alberto Jorge Olvera Salazar, B.Eng. (Chemical) (Universidad Autónoma de Nuevo León, NL. México), M.Sc. (Instituto Tecnológico y de Estudios Superiores de Monterrey, NL. México)

SUPERVISOR: Dr. T.E. Marlin

NUMBER OF PAGES: xi, 174

Abstract

In this thesis, a novel block decentralized MPC approach is implemented in order to coordinate the control of interacting process units (blocks) in a chemical plant. The goal of this research is to develop coordinated control that enables each block to optimize its own performance by adjusting only is manipulated variables while accounting for interactions among blocks.

A simultaneous algorithm, termed D-MPC, is proposed that replaces multiple optimizations (from several, interacting MPC controllers) with one set of equations, yielding a single-level optimization problem. Given the complexity of the resulting problem consisting of linear and complementarity equations, an efficient active set heuristic is proposed for real time computations. The approach is computationally tractable, yielding a small set of convex problems to be solved sequentially and providing reliable solutions with good dynamic performance for the cases studied.

Integrity is important for control designs, and generally, block designs with negative and zero Block Relative Gains (BRG) have poor integrity and cannot be controlled with published approaches. In contrast, the D-MPC approach successfully provides good integrity for processes with all BRG signs while maintaining the desired autonomy of each individual block.

The solution existence, uniqueness, and stability of the proposed controller are also discussed in order to delimit what kind of processes can be controlled using the proposed D-MPC controller. A simple D-MPC formulation is analyzed to demonstrate that specific ranges of controller tuning can lead to the loss of nominal stability for negative BRG systems. Therefore, a step-wise D-MPC design procedure was developed that integrates a stability analysis first proposed for centralized MPC and successfully adapted for the D-MPC controller.

The dynamic performance and integrity demonstrated in case studies with all signs of BRG and sizes from 2x2 to 4x4 demonstrate the computational tractability, good dynamic performance of D-MPC controller designs developed with the design procedure and implemented with the heuristic algorithm.

Acknowledgement

I would like to thank my wife Martha Ovalle de Olvera for her endless love and support and for giving me those beautiful kids, J. Alberto and Eric D., who are the motivation that keeps me going.

I would also like to thank my supervisor Dr. Thomas E. Marlin for his immense guidance, teaching and patience and especially for his continuing support during those difficult parts of this project.

I would like to thank my peers in the process control group for their support not only for academic matters. I would also like to thank all those beautiful people who gave us a hand and more at the beginning of our Canadian experience (Salvador, Matilde, Maritza, Enid, Danielle, Zhiwen).

I also want to thank the Department of Chemical Engineering, Conacyt-Mexico and the Canadian Bureau of International Education for their financial support.

Finally I want to thank my parents Jose Olvera and Isabel Salazar de Olvera for their continuing moral support and for all they have done for me.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Most industrial processes are formed by different unit operations interconnected by a set of process streams that can include streams and energy integration. This process integration introduces interacting effects that cannot be isolated to a part of the process. In the past, processes with recycle streams employed many surge tanks to buffer disturbances and minimize interaction (Luyben et al 1998). Even though this method slows interactions, it does not eliminate them, and it introduces additional capital and operating costs. Current design practice is to take full advantage of process integration without buffering and include advanced controls to provide adequate dynamic performance.

Historically, plantwide control was obtained by means of multiple SISO controllers usually in the form of PID (Proportional Integral Derivative) controllers. The strategy was to establish all the loops for each individual unit operation and then combine them together expecting that any conflict that may arise could be reconciled (Stephanopolous, 1984).

In the 1980's plantwide control was drastically improved by the successful application of centralized model predictive controllers (MPC). This type of controller optimizes the future trajectory of controlled variables using dynamic models that predict the effects of manipulated and measured disturbance inputs on the controlled variables. However, it is important to mention that there are not many published reports that discuss the application of MPC to an entire complex chemical plant. Instead, the current practice in process control of continuous operating process uses multiple block controllers (MPC) that do not consider interactions among blocks (Wagler, 2007; Jakhete et al, 1999). Figure 1.1 shows a sketch of the current practice of Block Decentralized MPC. (Here, we consider a block to be any combination of manipulated and controlled variables in a centralized block, so that even a single-input-single-output loop is a block.) A large MPC controller involving all plant controlled variables is not usually considered as an acceptable alternative, even if the computing power is available (Wagler, 2007).

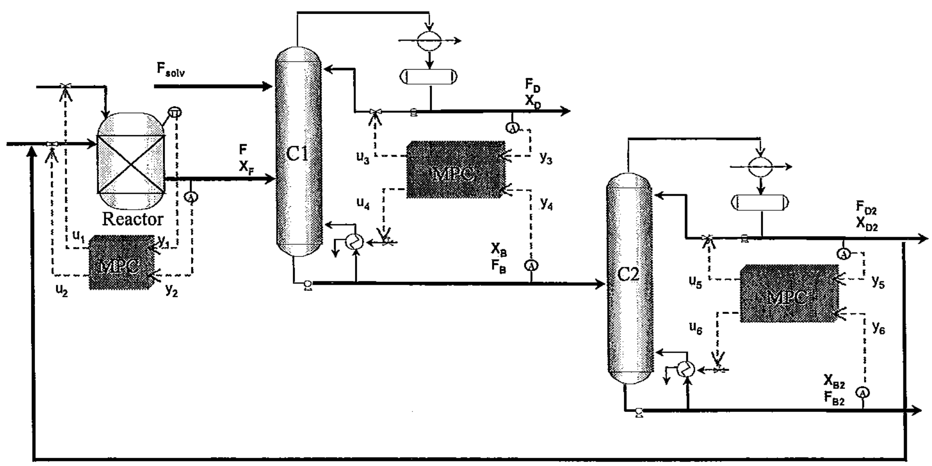


Figure 1.1 Local Control in Block Decentralized MPC

Based on the current state of industrial practice, this thesis deals with the *coordination* of multiple MPC controllers, which for the purposes of this thesis are defined as Block Decentralized MPC. Thus, the well-established MPC algorithm will be employed for each block, and the block controllers will be "coordinated" in a manner that will compensate for the deleterious effects of interactions among blocks. As we will see, this approach also expands successful application of block controllers to some processes that would otherwise be excluded.

1.1 Objective

The goal of this work is to develop a method for implementing block decentralized control based on the MPC algorithm for each block controller. To accommodate process interactions, the method will coordinate among the various blocks to provide good dynamic performance of the decentralized control systems.

Block Decentralized control could have one of two possible goals; (a) to emulate centralized control, or (b) to provide autonomy for each block controller. This work addresses the second, in which each controller achieves the best performance for its block, which is naturally subject to interactions for other blocks in the plant.

When considering block controllers, we must address two challenges; (a) the design of the blocks, which assigns measured controlled variables and manipulated variables to each block and (b) the real-time algorithm to implement block control. This research addresses the second challenge, while the first has been addressed in other research by, among others, Cai (2009).

It is important to note that this research *does not seek* to develop implementation approaches that (1) reduces the empirical modeling effort or (2) reduces real-time computations. As we will see, the modeling for decentralized MPC is the same as for centralized, because of the need to "coordinate" interactions. Also, the computations for block decentralized MPC must be acceptable for today's computing, not less than any other controller.

This thesis focuses on the development of a Novel Model Predictive Controller called **D-MPC,** which is used for the coordination of a set of Block Decentralized MPC Controllers. It addresses existence, stability, integrity, tuning and dynamic performance of the proposed controller.

1.2 Reason to Implement a Block Decentralized MPC (D-MPC)

Industrial practice typically employs multiple single-loop controllers (which could be considered block centralized) or a few block MPC. However, today's computing enables us to implement fully centralized for many plants. Therefore, it is worthwhile to explicitly address the reasons for selecting block decentralized MPC for some implementations. This research considers the following motivations for maintaining the implementation of Block MPC over a plantwide centralized MPC.

Block Autonomy: Each block controller should be able to achieve the best performance *for that block*, without considering the effect on other blocks. This approach follows the management goals in many companies, even though a more centralized approach might provide a better (global) optimum. In addition, the integration of automation between individual companies due to tight integration of production and consumption (for example, utilities or one product being the downstream raw material) and vendor-managed inventories (where a supplying company manages the inventory of their products in the customers plant).

Disturbance Isolation: In some cases, the best performance in a single company involves maintaining the effects of a significant disturbance in one block of the plant, rather than using all manipulated variables to attenuate the disturbance. For example, it might be better maintain products in all other blocks (not directly affected by the disturbance) "on specification", with only some variables in the directly-affected block having a few products "off-specification".

Fault Tolerance: In the case of undetected sensor faults, the control system will make *incorrect* adjustments to essential all manipulated variables under management of the controller (because of interactions). Therefore, a larger sized block will lead to a greater immediate effect of the fault transmitted through the controller.

Management: As the number of variables in a block decentralized controller increases, the monitoring and diagnosis of the controller actions becomes more challenging. This important factor works against fully centralized MPC control.

Dynamic Performance: Naturally, good dynamic performance is important for disturbance rejection and set point changes. Generally, blocks contain highly interactive combinations of manipulated and controlled variables, with weak interactions among blocks. Again, the block design is not addressed in this research, but the products of this research should be able to function for essentially any block structure.

In addition to the previous characteristics the controller must guarantee the existence of a controller solution and closed loop stability.

1.3 Main Contributions

The major contributions of this thesis are summarized as follows:

1. Unconstrained D-MPC formulation: The D-MPC controller applies a strategy similar to that of multilevel optimization. Here, several optimization problems at a same level are replaced with their respective optimality conditions and solved simultaneously.

2. Constrained Block D-MPC: The basic unconstrained D-MPC is extended to solve the constrained cases. The objective of implementing this strategy is to enhance the D-MPC formulation in a way that removes the non-convexity of the resulting controller. The modified strategy consists in a systematic method that iteratively detects constraints violations and automatically incorporates required bounds (active set) into the controller formulation. The approach is computationally tractable (yielding a small set of convex problems to be solved sequentially) and provides reliable solutions with good dynamic performance for cases studied.

3. Dominant interactions and integrity: In conventional block decentralized control (without coordination), strong interactions can lead to designs with poor integrity, so that turning one controller off (on) can cause another controller to become unstable. The previously-published Block Relative Gain (BRG) provides integrity analysis in the same manner as RGA does in multiloop control. The controller developed in this research is able to stabilize systems with positive, negative, and zero BRG control structures, which extends the range of processes and block designs for which block MPC is possible.

4. Closed loop stability: Once the existence of the controller has been achieved the next step is to guarantee the stability. Perhaps surprisingly, inappropriate tuning parameters can yield an unstable D-MPC controller, even *without model mismatch.* The stability analysis involves the application of classical linear stability analysis for discrete systems. When these results demonstrate nominal stability for selected tuning, they also provide a certificate for the existence (non-singularity) of the controller calculation.

1.4 Thesis Overview

Chapter 2 – Literature review

This chapter begins with a description of Model Predictive Control and specifically the Quadratic Dynamic Matrix Control (QDMC) version considered for this thesis. Then concepts of communication and cooperation between different controllers are introduced, followed by the competing technologies that employ such concepts. Finally the concepts of optimization required for this work such as bilevel optimization, interior point methods and the active set strategy are briefly described. This thesis is based on the idea of multiple controllers that maintain local autonomy and integrity. Their description as well as the concept of block relative gain (BRG) is presented in this chapter.

Chapter 3 – A Novel Block Decentralized Model Predictive Control (D-MPC).

The methodology to develop the D-MPC controller is here presented, starting with a QDMC controller and extending to the coordinated D-MPC controller. Unconstrained and constrained formulations are presented as well the heuristic approach to remove the non-convexity in the constrained case. Finally, a modified D-MPC is formulated that is able to achieve performance between centralized and block decentralized MPC.

Chapter 4 – Controllability and Stability Analysis.

This chapter presents a controllability criterion that defines what kind of plants can be controlled using the D-MPC. Then, the stability analysis for nominal unconstrained D-MPC is introduced. The analysis demonstrates that some D-MPC designs with "reasonable" tuning are not nominally stable, a result that was not anticipated. A procedure is developed to determine whether a nominally stable D-MPC can be achieved.

Chapter 5 – Block D-MPC Performance (Case Studies).

In this chapter a *two by two* distillation column is used as the main test case to demonstrate the capabilities of the D-MPC controller. First, independent objectives are defined and different design configurations (positive, negative and zero BRG) are tested. Then the effect of tuning for stability is considered. A couple of extra cases with higher dimensions are also considered.

Chapter 6 – Contributions and Future Work.

Finally this chapter presents a summary of the findings of this work as well as conclusions based on the results achieved. Recommendations for future work are also presented.

Chapter 2

Technology Survey

In the previous chapter it was stated that industrial plants have many block decentralized decision-making systems for automatic control and optimization. Since these systems interact, the performance of the integrated system could deviate from the required dynamic performance. Therefore, a coordination scheme is sought that retains the desired block decentralized autonomous decision-making but accounts for interaction.

In this chapter a review of the different concepts and technologies employed for the development of the coordinated Block D-MPC controller is presented. Additionally, a review of the published literature on the coordination of MPC is presented. This review introduces the reader to the main research lines involved in the coordination of MPC controllers.

2.1 Model Predictive Control

This entire work is based on the use of Model Predictive Control as the control algorithm. MPC is the most widely used advanced controller in industry and it refers to a class of control algorithms where dynamic process models are used to predict and control a process. MPC is well suited for high performance control of constrained multivariable processes because explicit pairing of controlled variables (CV) and manipulated variables (MV) is not performed and constraints are directly embedded in the problem formulation (Qin and Badgwell, 2003, Nath and Alzein 2002).

Theoretically, we might have a single centralized MPC controller for each large process plant. However, this is not the case found in practice, especially in cases where each plant section (block) desires to maintain the absolute control over its decision variables. Usually multiple MPCs are employed for dynamic control. These controllers can be coordinated through a steady state optimizer (LP or QP), which finds a feasible final steady state but does not account for dynamics.

An important part of the MPC application is definitively the process model embedded in the algorithm. Chemical processes are inherently nonlinear; however the most common approach in the MPC design is to express the model equations in a linear form. These models may be generated by empirical identification, first principle equations or a combination of both. These models are usually employed in the form of linear time invariant models (LTI) and are very common in industry.

Problem (2.1) shows a typical formulation of an MPC controller for a continuous process, the objective usually seeks to minimize the error between the controlled variables and the reference trajectories obtained from the economic optimizer.

$$\begin{aligned}&\min_{\Delta u}\left\{\left\|y-y^{sp}\right\|_Q^2+\left\|\Delta u\right\|_R^2+\left\|u-u^{ref}\right\|_\alpha^2\right\}\\&s.t.\\&y=f(\Delta u,d,\hat{N})\\&y_{\min}\le y\le y_{\max}\\&u_{\min}\le u\le u_{\max}\end{aligned}\tag{2.1}$$

In this case a set of manipulated variables are adjusted to drive the process to the desired steady-state operating point without violating constraints. The manipulated variable values for the first time step are implemented in the plant, and the problem is repeated at each controller execution time.

The weights of the quadratic norms in (2.1) are used to modify or "tune" the dynamic behaviour of the system and to achieve a MPC that is robust in the presence of model mismatch. The process output, $y \in \Re^n$ is a function of input moves, $\Delta u \in \Re^m$ measured disturbances, d and the estimated unmeasured disturbance, $\hat{N}$. The input variable u is obtained by adding Δu to its current value.

From the numerous MPC technologies available this work makes use of the Dynamic Matrix Control (DMC) algorithm (Cutler and Ramaker, 1980), which is widely used in the process industry for unconstrained control. For constrained control this work considers the Quadratic Dynamic Matrix Control (QDMC) algorithm as presented by Garcia and Morshedi (1986). Some of the key features of both algorithms that made it suitable for this research are the following:

- Linear step response models for the plant.
- Quadratic performance objective
- Optimal input computed as the solution of a convex Quadratic Programming problem.

Maybe the most important feature of QDMC is the ease with which constraints are incorporated in the problem, which are a key part of this work. The QDMC methodology will be described in detail in Chapter 3.

2.2 Key Concepts

The plantwide control structure considered for this work is the so-called Block Decentralized Structure, which is a combination of several controllers that can be single- or multi-variable (Cai, 2009). In a chemical plant a typical structure contains multiple (Block)

MPC controllers; usually each of them is handled in a decentralized way regarding the other controllers. This research states and solves the coordination of this problem in a way that preserves the autonomy of the controllers for each plant block.

It is worth emphasizing that this "decentralization" is not aiming for the reduction of the computational effort as main objective, but enforcing the decentralized independent goals. Some concepts required to understand the goals of this research are described next.

In this thesis, the controller resulting from the coordination of different autonomous MPC controllers will be referred to as Block D-MPC or just D-MPC. In the same way and for the purposes of this thesis the conventional technology, which addresses block MPC without coordination is referred as independent block MPC.

2.2.1 Local Autonomy

One of the key concepts that drive this research is maintaining local autonomy of the different sections of the plant. Basically, this means that each plant section (block) desires the absolute control over its decision (manipulated) variables. Although the goals of each block are independent their behaviour is definitely not because of the process interactions among plant blocks. This situation suggests that the problem should not be addressed as a set of independent subproblems.

The autonomy of the controllers mentioned above is achieved when each MPC adjusts the manipulated variables of its own process unit to optimize its own objective function. In Figure 1.1 a typical portion of a chemical plant composed by a reactor and two distillation columns is shown. Here, each unit has a local MPC controller. In case a manipulated variable of column C1 loses control or gets saturated, the response of a plantwide centralized MPC controller would be to adjust a manipulated variable in column C2 leading to a loss in performance on that column. On the other hand, the proposed coordinated D-MPC scheme has the objective to avoid a controller adjusting one of its manipulated variables to control a controlled variable of a different block.

Maintaining this local autonomy in the coordination scheme preserves disturbance isolation and fault tolerance of the control system. These two characteristics are the main reasons why independent block MPC is widely used in practice, although current methods suffer from dynamic interaction.

2.2.2 Integrity and Block Relative Gain

Decentralized controllers are widely used in industry because of their simplicity. They are easier to understand and to implement. However, the performance of a decentralized controller can be poor in the presence of severe process interactions. The same situation occurs with Block MPC structures like the one shown in Figure 1.1.

In decentralized SISO control the concept of integrity becomes important. Basically the system has integrity if the system is stable without changing signs of any feedback controller gains once one of the loops is placed in manual. We definitely want integrity, which requires that the sign of the process gain ($\Delta y / \Delta u$) is independent of the on/off status of other controllers. Additionally, Campo and Morari (1994) developed a series of integrity definitions for closed-loop systems. For the purposes of this work the **Decentralized Integral Controllability (DIC)** definition is the most suitable. Basically DIC implies that each controller can be detuned or put in manual independently, and the remaining closed-loop system will remain stable. The main requirement for DIC is that the relative gains (RGA) of the process must be positive.

When dealing with block decentralized control structures, integrity is addressed through the use of Block Relative Gain (BRG) (Manousiouthakis et al., 1986). BRG is an extension of the classical RGA pairing method presented by Bristol (1966). The use of Block Relative Gain (BRG) concepts provides a guideline for selection of suitable block structures in block decentralized control systems. Similar to the previous case the necessary condition for integrity in block decentralized structures is to have a BRG with a positive determinant (Chiu and Arkun, 1990). A brief introduction of BRG can be found in Appendix B.

2.2.3 Dynamic Performance

A major objective of the proposed Block D-MPC controller is to improve the dynamic performance of block decentralized structures. In this research it is a premise that the dynamic performance obtained via Block D-MPC coordination should be better than the solution obtained from an independent block MPC control system. Dynamic performance is a very rich concept and a single metric is never enough to measure it. Ideally a closed loop system must satisfy the following performance criteria (e.g., Seborg et al, 2003):

1. Closed Loop System must be stable.
2. Provide good disturbance rejection and set point tracking.
3. Steady-state error (offset) eliminated.
4. Excessive control action is avoided.
5. Must be insensitive to change in process conditions (Robust control).

Most of these performance criterions are embedded in the MPC objective function. For example the deviation of the controlled variable from the set point, the excessive control action as well as the offset requirement are enforced in the first two terms of the objective function as described in (2.1).

For a given control structure the problem of achieving good dynamic performance is reduced to the tuning problem. Shridhar and Cooper (1998) derived an analytical expression that computes appropriate move suppression factors for multivariable MPC, and it is based on keeping a low target value of the condition number of the system dynamic matrix. The optimal tuning of MPC controllers however, is a challenging problem, and it requires an explicit evaluation of the dynamic transient. Recently, Cai (2009) developed a combinatorial-based methodology to select both the optimal control structure and optimal tuning for independent block MPC designs. However, probably the fairest and most reliable way to obtain the best tuning of the controller would require evaluating the dynamic performance offering some robust stability.

Robust MPC is another active research area that considers several types of uncertainty (i.e. parametric, structural). To tackle this problem several methodologies have been developed, Warren (2004) proposed a chance constraint method where the uncertainty in the closed-loop model is characterized by an ellipsoidal uncertainty description. Parametric programming has also been used for Robust MPC (Kakalis et al, 2004).

Both Robust MPC and the optimal structure/tuning of MPC controllers are beyond the scope of this research. The proposed D-MPC controller will make use of trial-and-error tuning in order to satisfy the first four criterions described above.

2.3 Decentralized (Block) MPC

Coordination of Decentralized MPC control has received attention in recent years. In industry some of the most common methods ignore dynamic interactions among subsystems. Jakhete et al, (1999) published an industrial application where two processing units in a refinery, a fluid catalytic cracking unit (FCCU) and a gas processing unit (GPU) were coordinated. Although the main objective was the coordination of two Block MPC controllers, one for each unit, their implementation completely ignores the dynamic interactions and focused only on the steady state target coordination.

In this review two major categories are used to group the several Block Decentralized MPC approaches found in literature. The first category aims to solve the coordination of different controllers each of them having an independent objective function. The second category of approaches intends to match or approximate the centralized MPC performance by optimizing an objective function that is similar to the centralized MPC objective function.

The first category has been named as Communication MPC and incorporates a Block Design strategy. Here, each controller communicates part of its knowledge to other controllers by exchanging predictions for their local states and control moves. The second category is called Cooperation MPC, and it enforces a Centralized Design. In these formulations, the computations are decentralized among the individual MPC controllers, which cooperate towards achieving performance close to a conventional centralized MPC. Generally, the goal is to minimize computational effort or allow the computation to be distributed to several local processors. Figure 2.1 depicts several methods that fall under these two categories. Some of the most important works are briefly described in the next sections.

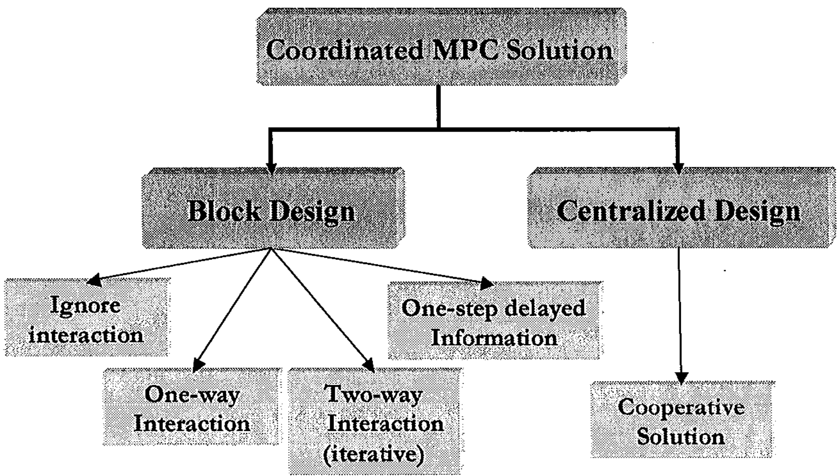


Figure 2.1 Technologies in Coordinated MPC

The work in this thesis has no intention to obtain a centralized performance; therefore it falls under the Block Design category and could be deemed as a Communication MPC approach.

2.3.1 Game Theory and the Concept Nash Equilibrium

Several authors have mentioned that the coordination of decentralized control is based on concepts from Game Theory such as Nash Equilibrium (NE) (Başar and Olsder, 1982; Van Shuppen, 2000; Negenborn et al 2004). Although game theory provides descriptive concepts, it not always tells us how to compute solutions. The characteristics of the decentralized MPC problem addressed in this research suggest that it may be classified as and *Extensive Game with perfect information and simultaneous moves* (Osbourne, 1994).

However not much insight other than the concept of Nash equilibrium can be obtained from the theory behind these games.

Negenborn et al (2004) described the MPC controller as an agent that has abilities to act and communicate with other agents to solve a given problem. Each agent has a reaction set that contains all the possible actions that an agent would make when it knows what the other agents will do. Two possible solutions to the multiagent problem are identified. (1) The Nash Equilibrium point identified as the intersection of the reaction sets of all agents. (2) The Pareto optimal solution included in the Pareto Set of solutions, which is the set of all feasible solutions to the overall problem. These two solutions for two, one-variable MPC controllers with interaction are depicted in Figure 2.2. The dashed ellipses represent the objective contours for each independent controller, with the other controller off. The dash-dot lines are to locations of the optimum for one controller, with the move by the other controller known.

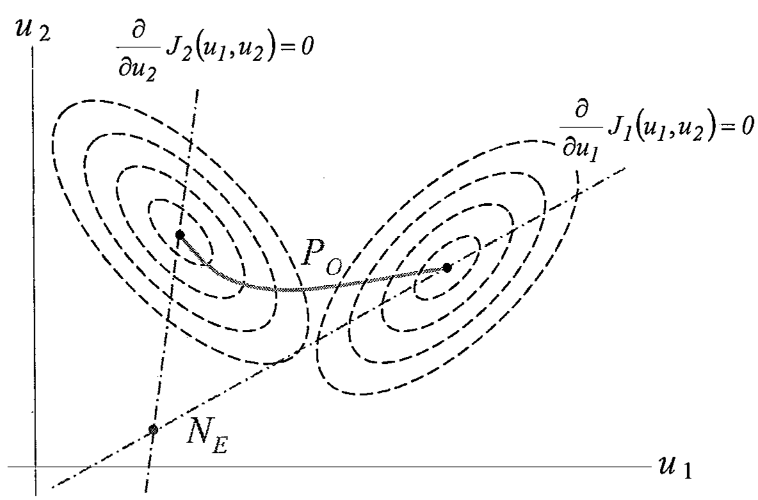


Figure 2.2 Nash and Pareto Optimal Sets (Giovanini, 2007)

Here it is important to mention a couple of observations:

(1) The elements of the Pareto set represent possible trade-offs among the multiple objectives of the subproblems.

(2) The solution of the overall centralized problem and the decentralized control problems are two extremes. These solutions correspond to the Pareto optimal (P_O) and the Nash Equilibrium solution (N_E), respectively.

For the purposes of this work a group of control decision $U^{N_E}(k)=\left(u_1^{N_E}, u_2^{N_E}, \cdots u_N^{N_E}\right)$ is called to be Nash optimal solution if the following relation is held (Du et al. 2001).

$$J_i\left(u_1^{N_E}, \cdots, u_i^{N_E}, \cdots u_N^{N_E}\right) \leq J_i\left(u_1^{N_E}, \cdots, u_{i-1}^{N_E}, u_i, u_{i+1}^{N_E} \cdots u_N^{N_E}\right), \quad \forall u_i, \quad i=1,2,\cdots N \tag{2.2}$$

When the Nash solution is achieved, each agent (i) has achieved the local optimum; any change in its control decision (u_i) will degrade the local performance index (J_i).

Once the theoretical concepts of Nash Equilibrium and Pareto Optimum have been introduced the next step is to present the different approaches available in literature that address the decentralized MPC problem.

2.3.2 Block Design - Communication MPC

Communication MPC methods focuses on MPC controllers for individual blocks integrated with an approach to handle interactions among blocks. The solution sought in this approaches is that of a Nash equilibrium point.

Maybe one of the first works that addressed the decentralized MPC problem is from Charos and Arkun (1993). In this work the authors proposed a new formulation of the QDMC controller based on the decomposition of the original problem into smaller subproblems, which then can be solved sequentially. The formulation relies in the fundamental assumption that each other controller will keep the already implemented inputs constant for the next prediction horizon. The approach uses a sequential solution of MPC problems, with each subsequent controller having knowledge of the previously calculated optimization results. Since this approach does not include multiple iterations, most control calculations do not have full knowledge of other block results, and the dynamic performance could be poor.

Later Jia and Krogh (2001) and Camponogara et al. (2002) studied state–space based distributed MPC formulations and their stability characteristics. They presented a Distributed MPC algorithm where each subsystem applies a local MPC. The local state predictions are communicated to other subsystems and incorporated in the control calculations. This approach is similar to the one proposed by Charos and Arkun (1993), in the sense that interactions among processes are considered based on a similar assumption (a one-step delayed exchange prediction). The controllers are solved sequentially using the state predictions from other subsystems evaluated with information from the previous execution time.

These previous methods can be classified as one-way interaction strategies where some kind of feed forward control is implemented. The next set of formulations considers two-way interactions and proposes some strategies to obtain the solution of the problem.

Du et al. (2001) and later Li et al (2005) in the same group studied the input coupling among step-response models for distributed MPC solutions based on Nash optimality. In this formulation, the controller calculations are performed iteratively, until convergence is obtained. In addition, these studies include a stability analysis. However, only unconstrained MPC is considered. An important shortcoming in this work is that the proposed iterative algorithm is only effective for systems with diagonal dominant gain matrix narrowing its application to positive BRG configurations.

Another formulation presented by Shigueo and Hong (2005) introduced a methodology based in the formulation of Charos and Arkun (1993) for QDMC. The procedure consists of an iterative algorithm where the key idea is to make use of available future prediction in order to calculate the next control actions. There is no proof that this method will converge to global optima or even to a local solution. However the "warm start" strategy seems to have a positive impact in the solution strategy.

More recently Al-Gherwi et al (2008) proposed a method to assess the robustness of communication based MPC. Basically it finds the control structure that better handles model errors by minimizing a variability metric trough input weights manipulation.

2.3.3 Centralized Design - Cooperation MPC

Cooperation MPC consists of a decentralized control structure where each MPC controller has a global objective function. The objective is to achieve "close" to a centralized performance. The cooperation MPC approaches are based on the assumption that computing the centralized solution may not be practical or reliable for large systems. These methods emerge to solve the sub-optimality (in the plantwide sense) obtained with communication-based MPC.

Venkat et al (2004) presented a cooperation-based MPC approach that challenged the existing communication MPC formulations from a stability and optimality perspective. They claimed that the Nash Equilibrium solutions are unstable and proposed a formulation where the local MPC objectives were replaced with global performance measures. With this modification each controller minimizes a projected objective involving all plantwide variables. A weighted sum of the local objective function is proposed for plantwide objective. It is important to mention that if the objective function of every subsystem is the same (i.e. Quadratic) the global objective matches exactly a centralized quadratic objective function. Therefore, the solution could be similar to that of a plantwide Centralized MPC.

Regarding the solution strategy, this is the same as in the communication MPC approach. Thus an iterative algorithm is implemented to find the optimal solution. Because the iterative procedure may not converge during the execution time, some modifications are made in order to guarantee feasibility of the method even if the algorithm does not converge at all. Another characteristic is the inclusion of terminal constraints to obtain closed-loop nominal stability for open-loop unstable plants.

Recently, Zhang and Li (2007) presented an approach called Network MPC, which is very similar to the cooperation MPC described above. It basically makes use of a centralized objective function that it is optimized in each local controller.

2.3.4 Decentralized LP-MPC

Almost all industrial MPC controllers consider an upper level steady-state optimizer, which usually consists of a target calculation problem posed as a linear programming (LP) or quadratic programming (QP) problem. The inclusion of upper-level problem maintains the feasibility of the controlled actions. This local optimizer may serve either as an integrating level between the steady-state RTO and regulatory level.

Cheng and Forbes (2004) proposed an approach to address the steady-state coordination of MPC controllers focusing on the steady state target calculations. In this method the ultimate goal is to find a coordinating strategy that obtains the same solution as the centralized approach. They follow the Dantzing-Wolfe decomposition principle to coordinate the interactions among decentralized MPC blocks. The resulting master problem would be the same as that of a plantwide centralized problem. The ultimate goal of this method is to reduce the computational requirements without any consideration of the dynamic performance.

A follow up work presented by Cheng et al. (2007) proposed a price adjustment method to solve the coordination of the LP-MPC problem. In this case the objective is not necessarily obtaining a centralized performance but maintaining the decentralized independence of the process units while coordinating the interactions.

2.4 Optimization Technologies

This work considers different optimization technologies in order to approach the Decentralized MPC problem. The main goal is to overcome the limitations encountered in the current communication-based MPC formulations, which are due to the use of an iterative algorithm.

First, multilevel optimization is discussed in order to present the framework on which the proposed D-MPC controller is developed. A couple of optimization algorithms are then discussed: (1) Interior Point and (2) Active Set Strategy.

2.4.1 Multilevel Optimization

Multilevel optimization problems are structured in a hierarchical way, where the upper level is executed first, and the solution influences the objective and feasible set for the lower levels. This may be contrasted with decomposition techniques, where a single objective is used to describe all decisions.

For the sake of simplicity multilevel optimization can be described using a bilevel problem; mathematically the problem can be stated as follows.

$$\begin{aligned} &\max_{x \in \Re^{n_1}} \ P(x,y) \\ &s.t. \ \ g(x,y) \leq 0 \\ &\quad \min_{y \in \Re^{n_2}} \ f(x,y) \\ &\quad st. \ \ h(x,y) \leq 0 \end{aligned} \tag{2.3}$$

A common method to solve this bilevel problem is to transform the problem into a single-level optimization problem by substituting the lower-level problem with its first order Karush-Kuhn-Tucker (KKT) conditions. The resulting problem is the following.

$$
\begin{aligned}
&\max_{x,y,w} \; P(x,y) \\
&s.t. \;\; g(x,y) \le 0 \\
&\left.\begin{cases}
\nabla_y f(x,y) + w \cdot \nabla_y h(x,y) = 0 \\
h(x,y) \le 0 \\
w \cdot h(x,y) = 0 \\
w \ge 0
\end{cases}\right\} \leftarrow KKT
\end{aligned}
\tag{2.4}
$$

Even under suitable convexity assumptions of both levels, the above mathematical program is very difficult to solve, due mainly to the nonconvexities that occur in the complementarity and Lagrangian constraints (Clark and Westerberg, 1990). While the Lagrangian constraint is linear in certain important cases (linear or convex quadratic functions), the complementarity constraint is intrinsically non-convex, and it is best addressed by enumeration algorithms, such as branch-and-bound or more recently by Interior Point Methods (Colson et al, 2005).

This methodology that replaces an optimization problem by its KKT conditions is the central part of the proposed D-MPC controller. As stated above the solution of this multilevel problem deals with complementarity constraints that cannot be solved using conventional NLP methods, such as SQP.

2.4.2 Interior Point Methods (IPM) and Complementarity Constraints

Interior Point or Barrier Methods are optimization methods that transform a constrained problem into a series of unconstrained ones. These methods follow a barrier approach where the inequality constraints are replaced by a barrier term (i.e. logarithmic function) that is added to the objective function.

Consider an Original NLP optimization problem:

$$\begin{aligned} &\min_{x \in \Re^n} f(x) \\ &s.t.\ c(x) = 0 \\ &x \geq 0 \end{aligned} \tag{2.5}$$

Using a barrier term to replace the bounds the problem is transformed into the following Interior Point (IP) problem:

$$\begin{aligned} &\min_{x \in \Re^n} \varphi(x) = f(x) - \mu \sum_i \log(x^{(i)}) \\ &s.t.\ c(x) = 0 \end{aligned} \tag{2.6}$$

For the resulting problem the objective function becomes arbitrarily large as x approaches the boundary defined by the inequality constraints, therefore the local solution of this problem lies in the interior of the constraints set. Then as μ approaches to zero the solution of the interior point (IP) problem approaches the optimal solution of the original problem. The strategy for solving the original NLP is to solve a sequence of barrier problems for a decreasing parameter μ.

The complexity of the problem increases when complementarity conditions are introduced (i.e. multilevel optimization). For these cases the optimization problem is the following:

$$\begin{aligned} &\min_{x \in \Re^n, w \in \Re^m, y \in \Re^m} f(x, w, y) \\ &st. \\ &c(x, w, y) = 0 \\ &x, w, y \geq 0 \\ &w^{(i)} y^{(i)} = 0,\ i = 1 \ldots m \end{aligned} \tag{2.7}$$

Where $w^{(i)} y^{(i)} = 0$ are the complementarity constraints. Then, by applying the barrier terms the original problem is transformed in the following IP problem:

$$\begin{aligned}
&\min_{x\in\Re^n, w\in\Re^m, y\in\Re^m} f(x,w,y) \\
&-\mu\left(\sum_{i=1}^{n}\ln\left(x^{(i)}\right)+\sum_{i=1}^{m}\ln\left(w^{(i)}\right)+\sum_{i=1}^{m}\ln\left(y^{(i)}\right)\right) \\
&st. \\
&c(x,w,y)=0 \\
&w^{(i)}y^{(i)}=0,\quad i=1\ldots m
\end{aligned} \tag{2.8}$$

As mentioned above this problem is highly non-convex, thus difficult to solve. Recently Raghunathan and Biegler (2003) developed an algorithm called IPOPT-C able to handle optimization problems with equilibrium constraints (MPEC). The main strategy is to relax the complementarity constraints $w^{(i)}y^{(i)}=0$ in the following way.

$$\begin{aligned}
&w^{(i)}y^{(i)}+s^{(i)}=\delta\mu \\
&s^{(i)},w^{(i)},y^{(i)}\geq 0
\end{aligned} \tag{2.9}$$

Where μ is the positive barrier parameter that is progressively reduced to zero. More recently, Baker (2006) reported that this method performed better than the most common NLP solvers such as CONOPT and MINOS.

2.4.3 Active Set Strategy

Quadratic Programming (QP) methods are widely used in process control applications. For example the QDMC algorithm used in this research is a QP problem where the (output) constraints are represented by the linear dynamic model. The solution method behind most QP solvers is based on the selection of working sets of active constraints, hence the name active set strategy.

Active set solvers for quadratic programming commonly consist of two phases. In Phase I a first calculation is performed to find an initial feasible point. Then, in Phase II the KKT matrix is updated as constraints are added or dropped while the algorithm reduces the objective function and maintains feasibility (Bartlett and Biegler, 2006).

The general formulation of a QP problem is:

$$\begin{aligned} &\min_{x \in \Re^n} f(x) = g^T x + \frac{1}{2} x^T G x \\ &s.t. \quad \begin{bmatrix} x_L \\ c_L \end{bmatrix} \leq \begin{bmatrix} I \\ A_C{}^T \end{bmatrix} x \leq \begin{bmatrix} x_U \\ c_U \end{bmatrix} \end{aligned} \tag{2.10}$$

Where $x, x_L, x_U \in \Re^n$, $g \in \Re^n$, $G = G^T \in \Re^{nxn}$ is positive definite, $A_C \in \Re^{nxm}$ and $c_L, c_U \in \Re^m$. The QP algorithm attempts to solve the first-order KKT optimality conditions, which include the feasibility of the constraints and the stationary constraint.

$$g + Gx + A_C + \lambda + \mu = 0 \tag{2.11}$$

The solution consists of the primal variables x and the multipliers $v^T = \lfloor \lambda^T \quad \mu^T \rfloor$. At the optimum the elements of v will satisfy the complementarity conditions: $v \geq 0$ for an active upper bound, $v \leq 0$ for an active lower bound, and $v = 0$ for an inactive constraint.

An active-set strategy searches through different working sets of active constraints until the optimality conditions are met. At every working set (iteration) the problem (2.10) is transformed into an optimization problem with equality constraints.

$$\begin{aligned} &\min_{x \in \Re^n} f(x) = g^T x + \frac{1}{2} x^T G x \\ &s.t. \quad A_{ws}{}^T x = b_{ws} \end{aligned} \tag{2.12}$$

Where A_{ws} contains the information of the active constraints and b_{ws} is either the upper or level bound for the working set. The methodology used to select which constraint is added or dropped is basically what distinguish the different solvers available.

Later in Chapter 3 a heuristic method based on the concepts described in this section is proposed to solve the constrained version of the decentralized MPC problem.

Chapter 3

A Novel Block Decentralized MPC (D-MPC)

In this chapter the Block D-MPC controller is presented in detail. Recall that the goal of the D-MPC is to provide block-decentralized control with autonomy within blocks and good dynamic performance of all blocks. Here, the D-MPC is formulated for unconstrained and constrained cases. A heuristic method based on an active set strategy is introduced as a way to remove the nonconvexity of the coordination problem and facilitate real-time computation. Finally an extended controller capable of achieving either a D-MPC or a fully centralized (conventional) performance is presented.

3.1 Unconstrained D-MPC Formulation

This section describes the formulation of the Block Decentralized MPC (D-MPC). The Unconstrained D-MPC formulation begins by formulating each MPC controller as a MIMO DMC controller as described in Cutler and Ramaker (1980). This formulation is based on the standard MPC technology presented in Appendix A.

The D-MPC formulation considers two or more MPC controllers to be coordinated to meet the specified goals. These controllers and their associated processes are referred to as blocks throughout this thesis. A block may encompass either a MIMO or a SISO process, with the manipulated and controlled variables defining each block. In the first chapter Figure 1.1 showed a schematic of a typical D-MPC problem.

A couple of important assumptions that are used in this thesis are now made.

1. Every controller has the **same** prediction (p) and control horizon (m), which is not required but done consistently in this work for ease of notation.

2. Every controller must have the **same** execution period. (This is not strictly necessary; controllers could have execution periods that are integer multiples, which would complicate the programming, tuning, etc.)

Following the basic formulation for Dynamic Matrix Control, the MPC controller in block i has the following formulation.

$$\underset{\Delta u_i}{Min}\ J_i = \frac{1}{2}\cdot\left(\left\|y_i - y_i^{sp}\right\|_{Q_i}^2 + \left\|\Delta u_i\right\|_{R_i}^2\right) \tag{3.1}$$

$$y_i = A_{ii}\Delta u_i + E_i + y_i^P + d_i \tag{3.2}$$

$$E_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} A_{ij}\cdot\Delta u_j \tag{3.3}$$

where

$$\Delta u_i \in \Re^{(m_i\cdot M_i)}$$

$$y_i, y_i^{sp}, y_i^P, d_i, \in \Re^{(p_i\cdot P_i)}$$

$$A_{ii} \in \Re^{(p_i\cdot P_i)\times(m_i\cdot M_i)}$$

$$A_{ij} \in \Re^{(p_i\cdot P_i)\times(m_j\cdot M_j)}\quad j\neq i$$

$$Q_i \in \Re^{(p_i\cdot P_i)\times(p_i\cdot P_i)}$$

$$R_i \in \Re^{(m_i\cdot M_i)\times(m_i\cdot M_i)}$$

Equation (3.2) represents the linear dynamic model; the term A_{ii} contains either a SISO or a MIMO dynamic matrix that describes the effect of inputs in block i to outputs in block i. The structure of A_{ii} has been described in Appendix A. The term E_i described in (3.3) contains the effect from inputs in blocks $j \neq i$ to outputs in block i, which represents the interaction among processes; changes to the manipulated variables in blocks $j \neq i$ are measured disturbances for the block i controller.

Parameters p_i and m_i are the prediction and control horizon of block i respectively. In the same way P_i and M_i are the number of process output and input variables of block i. The previous assumptions indicate that the prediction and input horizons are the same for every controlled block, therefore p_i and m_i can be represented as p and m.

It is important to emphasise that the optimization variables are Δu_i, which consists of the control actions of only the input variables defined for block i.

The effect on the predicted output y_i from past inputs Δu_i and Δu_j $(j \neq i)$ is included in the vector y_i^p. The output feedback is the vector d_i and corresponds to the "model error", which includes effects of unmeasured disturbances and model mismatch. This vector d_i can be calculated in different ways, the most common method, which is used in this thesis, is the difference between the measured and predicted values of y_i at the time of the controller execution, which is assumed constant throughout the entire horizon.

It is important to note that the structure of the vectors is properly handled by stacking multiple variables. For example for a specific block i variables are handled as follows.

$$\Delta u_i = \begin{bmatrix} \Delta u_{i1,(k)} \\ \vdots \\ \Delta u_{i1,(k+m-1)} \\ \hdashline \Delta u_{i2,(k)} \\ \vdots \\ \Delta u_{i2,(k+m-1)} \\ \hdashline \vdots \\ \hdashline \Delta u_{iM_i,(k)} \\ \vdots \\ \Delta u_{iM_i,(k+m-1)} \end{bmatrix} \qquad y_i = \begin{bmatrix} y_{i1,(k+1)} \\ \vdots \\ y_{i1,(k+p)} \\ \hdashline y_{i2,(k+1)} \\ \vdots \\ y_{i2,(k+p)} \\ \hdashline \vdots \\ \hdashline y_{iP_i,(k+1)} \\ \vdots \\ y_{iP_i,(k+p)} \end{bmatrix}$$

Finally matrices Q_i and R_i are the block tuning parameters and in this formulation Q_i is defined as diagonal and positive definite matrix (PD) and R_i is defined as diagonal positive semidefinite (PSD) or definite (PD) matrix. These tuning parameters are deemed local because they only account for input and output variables of block i.

3.1.1 Control Calculation for Each Individual Block (QDMC formulation)

Looking ahead, we recognize that we will be solving the optimization for multiple blocks simultaneously. To avoid a multi-level optimization problem that would be intractable, we will reformulate the individual block optimizations so that multiple blocks can be optimized in a tractable manner. The first step in this formulation is the transformation of the optimization problem into a well-posed problem for real-time solution. An important part is to explicitly express the interaction effects.

Expanding the first term of (3.1).

$$\left\| y_i - y_i^{sp} \right\|_{Q_i}^2 = \Delta u_i^T \left(A_{ii}^T Q_i A_{ii} \right) \Delta u_i + 2\left[A_{ii}^T Q_i \left(e_i + E_i \right) \right] \Delta u_i + \left(e_i + E_i \right)^T Q_i \left(e_i + E_i \right) \tag{3.4}$$

Where.

$$e_i = y_i^p + d_i - y_i^{sp}$$

It is important to mention that e_i contains the feedback information effects of past control outputs and forecast of the future set points and is not influenced by future control decisions.

The second term is easily transformed.

$$\|\Delta u_i\|_{R_i}^2 = \Delta u_i^T R_i \, \Delta u_i \tag{3.5}$$

Finally the objective function J_i can be expressed as.

$$\underset{\Delta u_i}{Min}\ J_i = \frac{1}{2}\Delta u_i^T \left(A_{ii}^T Q_i A_{ii} + R_i\right)\Delta u_i + \left[A_{ii}^T Q_i \left(e_i + E_i\right)\right] \Delta u_i + \frac{1}{2}\left(e_i + E_i\right)^T Q_i \left(e_i + E_i\right) \tag{3.6}$$

The resulting problem is an unconstrained QP of the following form.

$$\underset{x}{Min}\ \varphi(x) = x^T \Sigma x + c^T x + h$$

The solution of this unconstrained optimization problem must satisfy the following stationary condition.

$$\frac{dJ_i}{d\Delta u_i} = \left(A_{ii}^T Q_i A_{ii} + R_i\right)\Delta u_i + A_{ii}^T Q_i \left(E_i\right) + A_{ii}^T Q_i \left(e_i\right) = 0 \tag{3.7}$$

The second order condition basically requires that in order to have a convex optimization problem, the Hessian matrix $\Sigma = \left(A_{ii}^T Q_i A_{ii} + R_i\right)$ must be at least positive semidefinite, which is the case for a typical well-posed MPC problem. Therefore, a local minimum is a global minimum.

Note that.

- We will require that the inverse of the Hessian (Σ) in equation (3.7) exist through selection of appropriate process applications and values of tuning parameters. This topic is developed in the next chapter.

- If there is no interaction, the problem simplifies to a single loop MPC where the solution is the following.

$$\Delta u_i = \left(A_{ii}{}^T Q_i A_{ii} + R_i\right)^{-1} A_{ii}{}^T Q_i\left(-e_i\right) \tag{3.8}$$

- This set of linear equations can be solved for the optimization values of future adjustments of the block-manipulated variables.

- If this optimization problem is solved independent of other block controllers, the term E_i is treated as a zero (no coordination) or as a constant term (sequential coordination).

- The term e_i contains the feedback information and the effect from past inputs.

3.2 Simultaneous Solution for the Unconstrained D-MPC

The proposed approach for the coordination of Decentralized Block MPC controllers starts with the MPC formulation of each of the blocks that will be coordinated, i.e., equation (3.8). We note that this control problem is unconstrained, so that bounds on manipulated variables will be addressed in the subsequent section. With all controllers solved simultaneously, the communication among blocks will be through the interaction terms. These controllers are designed in a way that only the blocks' input variables (Δu_i) are the optimization variables for that block's controller, i.e., are the variables adjusted to minimize the blocks controller objective function.

The optimality conditions for all block controllers take the same form and can be combined and solved simultaneously as a set of linear equations, as given in the following.

$$\left(A_{ii}{}^T Q_i A_{ii} + R_i\right)\Delta u_i + A_{ii}{}^T Q_i\left(E_i\right) + A_{ii}{}^T Q_i\left(e_i\right) = 0 \qquad i = 1,\cdots,N \tag{3.9}$$

The interactive term in equation (3.9) is given in the following expression.

$$A_{ii}{}^{T}Q_i(E_i)=A_{ii}{}^{T}Q_i\sum_{\substack{j=1\\ i\neq j}}^{N}A_{ij}\Delta u_j \tag{3.10}$$

Expanding the interactive term for system ($i=1$) yields.

$$A_{11}{}^{T}Q_1(E_1)=\left(A_{11}{}^{T}Q_1A_{12}\right)\cdot\Delta u_2+\left(A_{11}{}^{T}Q_1A_{13}\right)\cdot\Delta u_3+\cdots+\left(A_{11}{}^{T}Q_1A_{1N}\right)\cdot\Delta u_N \tag{3.11}$$

Repeating for $i=2,\cdots,N$ results in equation (3.9) being rearranged to a set of linear equations.

$$\begin{bmatrix} A_{11}{}^{T}Q_1A_{11}+R_1 & A_{11}{}^{T}Q_1A_{12} & \cdots & \cdots & A_{11}{}^{T}Q_1A_{1N} \\ A_{22}{}^{T}Q_2A_{21} & A_{22}{}^{T}Q_2A_{22}+R_2 & \vdots & \vdots & A_{22}{}^{T}Q_2A_{2N} \\ \vdots & \cdots & \ddots & \vdots & \vdots \\ \vdots & \cdots & \cdots & \ddots & \vdots \\ A_{NN}{}^{T}Q_NA_{N1} & A_{NN}{}^{T}Q_NA_{N2} & \cdots & \cdots & A_{NN}{}^{T}Q_NA_{NN}+R_N \end{bmatrix}\cdot\begin{bmatrix}\Delta u_1\\ \Delta u_2\\ \vdots\\ \vdots\\ \Delta u_N\end{bmatrix}+\begin{bmatrix}A_{11}{}^{T}Q_1\cdot e_1\\ A_{22}{}^{T}Q_2\cdot e_2\\ \vdots\\ \vdots\\ A_{NN}{}^{T}Q_N\cdot e_N\end{bmatrix}=\begin{bmatrix}0\\0\\ \vdots\\ \vdots\\ 0\end{bmatrix} \tag{3.12}$$

Then, defining the following matrices.

$$A_C=\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N}\\ A_{21} & A_{22} & \cdots & A_{2N}\\ \cdots & \cdots & \ddots & \cdots\\ A_{N1} & \cdots & \cdots & A_{NN}\end{bmatrix}\quad Q=\begin{bmatrix} Q_1 & 0 & 0 & 0\\ 0 & Q_2 & 0 & 0\\ 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & Q_N\end{bmatrix}$$

$$R=\begin{bmatrix} R_1 & 0 & 0 & 0\\ 0 & R_2 & 0 & 0\\ 0 & 0 & \ddots & 0\\ 0 & 0 & 0 & R_N\end{bmatrix}\quad e=\begin{bmatrix}e_1\\ e_2\\ \vdots\\ e_N\end{bmatrix} \tag{3.13}$$

Equation (3.12) can be also expressed in a condensed form as follows.

$$\left(A_D{}^{T}Q\,A_C+R\right)\Delta u+A_D{}^{T}Q\cdot(e)=0 \tag{3.14}$$

Where A_D contains only the diagonal blocks (A_{ii}) elements of A_C.

$$A_D = \begin{bmatrix} A_{11} & 0 & 0 & 0 \\ 0 & A_{22} & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & A_{NN} \end{bmatrix} \tag{3.15}$$

And the decision variables of all blocks are in the stacked vector defined as $\Delta u = \begin{bmatrix} \Delta u_1^T & \Delta u_2^T & \cdots & \Delta u_N^T \end{bmatrix}^T$.

In conclusion the simultaneous set of equations for the unconstrained D-MPC can be expressed in the following way.

$$A_{DMPC}\, \Delta u = b_{DMPC}$$

Where

$$A_{DMPC} = \left(A_D^T Q\, A_C + R\right) \tag{3.16}$$

$$b_{DMPC} = A_D^T Q(-e)$$

Finally, the controller actions are calculated by using the following expression.

$$\Delta u = \left(A_D^T Q\, A_C + R\right)^{-1} A_D^T Q\, (-e) \tag{3.17}$$

The resulting **Block D-MPC** has a similar structure to that of the Centralized MPC as shown below.

C-MPC $$\Delta u = \left(A_C^T Q\, A_C + R\right)^{-1} A_C^T Q\, (-e)$$

The differences between C-MPC and D-MPC are obviously a result of designing the D-MPC controller to achieve different performance goals.

A geometrical interpretation of this problem is shown schematically in Figure 3.1 for a two-dimension problem. The solution obtained by the D-MPC controller is derived by allowing each controller to optimize its own objective, without regard for other objectives. The approach is shown as the intersection of the optimality conditions (dashed lines) for the two individual controllers. This optimization approach reaches a solution that is commonly called as the Nash Equilibrium point (Van Shuppen, 2004), shown as N_E in the figure. Equation (3.17) calculates the intersection of the optimality conditions, in this case the solution of a linear set of equations in the Δu space.

On the other hand the Pareto optimal path is the set of points (u_1, u_2) obtained by a centralized controller adjusting both manipulated variables to minimize the weighted function $J = (w_1 J_1 + w_2 J_2)$ for each $0 \le w_1, w_2 \le 1$, $w_1 + w_2 = 1$. The curve describing the range of Pareto solutions is designated by P_O. In this way the Pareto optimal solution obtained when $w_1, w_2 = 0.5$ corresponds to the solution of the C-MPC using the common tuning of equal weighting of controlled variables.

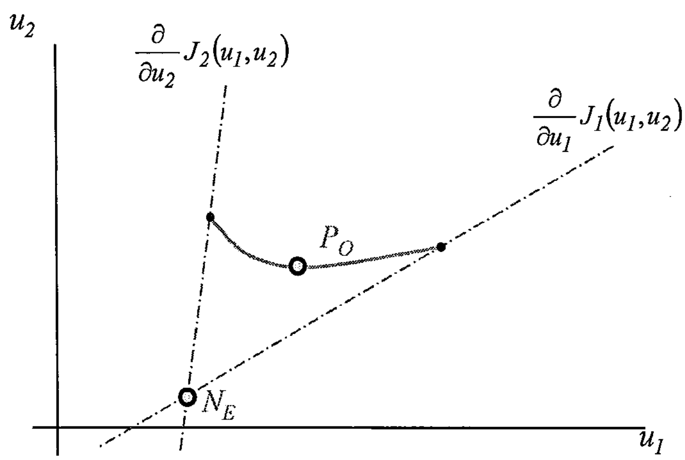


Figure 3.1 Nash (D-MPC) and Pareto Optimal (C-MPC) Solution

In this section it has been shown that when unconstrained QDMC controllers are considered for D-MPC, the solution is obtained from the stationary conditions, and the solution corresponds to solving a set of linear equations. This is the simplest situation, and no more than a linear solver is required for its real-time implementation.

3.3 Constrained Block D-MPC

In this section the control algorithm is extended to deal with the coordination of a set of MPC controllers with bounds on the manipulated variables. Again, the main approach for solving this problem is to replace each optimization problem (MPC) with its equivalent set of optimality conditions, thus yielding a single-level problem.

The QDMC formulation for the controller in block i, which incorporates hard input constraints, is the following.

$$\begin{aligned} &\underset{\Delta u_i}{Min}\ J_i = \frac{1}{2}\cdot\left(\left\|y_i - y_{sp_i}\right\|_{Q_i}^2 + \left\|\Delta u_i\right\|_{R_i}^2\right) \\ &\textit{subject to} \\ &\quad y_i = A_{ii}\Delta u_i + E_i + y_{p_i} + d_i \\ &\quad \Delta u_i^{\min} \le \Delta u_i \le \Delta u_i^{\max} \\ &\quad u_i^{\min} \le u_i \le u_i^{\max} \end{aligned} \tag{3.18}$$

where

$$E_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} A_{i,j}\cdot\Delta u_j$$

This formulation maintains the same structure as in problem (3.1) with the same dimensions for all arrays and the addition of input bounds $u_i^{\min}, u_i^{\max}, \Delta u_i^{\min}, \Delta u_i^{\max} \in \Re^{(m\cdot M_i)}$.

3.3.1 Input Constraints

Following the procedure used in Section 3.1.1 the next step is to represent the constraints in a suitable matrix form. First, the f^{th} input constraints in block i, $u_{i_f}^{\min} \le u_{i_f} \le u_{i_f}^{\max}$ can be described as follows.

$$u_{i_f}^{\min} \le u_{i_{f,(k)}} + \sum_{j=0}^{\mu} \Delta u_{i_{f,(k+j)}} \le u_{i_f}^{\max}$$
$$for \;\; \mu = 1,2\cdots,m-1 \tag{3.19}$$

Where $u_{i_{f,(k)}}$ is the current value of u_{i_f}. This constraint can be expressed in matrix form as.

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} (u_{i_f}^{\min} - u_{i_{f,(k)}}) \le \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdot & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta u_{i_{f,(k)}} \\ \Delta u_{i_{f,(k+1)}} \\ \vdots \\ \vdots \\ \Delta u_{i_{f,(k+m-1)}} \end{bmatrix} \le \begin{bmatrix} 1 \\ 1 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} (u_{i_f}^{\max} - u_{i_{f,(k)}}) \tag{3.20}$$

Throughout this section, ν denotes the vector of appropriate dimension of all ones. Then, defining $V \in \Re^{m_i \times m_i}$ as follows.

$$V = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdot & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.21}$$

The input constraint can be expressed as the following.

$$\begin{bmatrix} V \\ -V \end{bmatrix} \cdot \Delta u_{i_f} + \begin{bmatrix} \nu \cdot (u^{\max}{}_{i_f} - u_{i_{f,(k)}}) \\ -\nu \cdot (u_{i_{f,(k)}} - u^{\min}{}_{i_f}) \end{bmatrix} \le 0 \tag{3.22}$$

In a further condensed way the constraints for the f^{th} input variable in the i^{th} block are the following.

$$V_B \cdot \Delta u_{i_f} + \gamma_{i_f} \leq 0 \tag{3.23}$$

Where $V_B \in \Re^{2m \times m}$ and $\gamma_{i_f} = \begin{bmatrix} \nu \cdot (u^{\max}{}_{i_f} - u_{i_{f,(k)}}) \\ -\nu \cdot (u_{i_{f,(k)}} - u^{\min}{}_{i_f}) \end{bmatrix} \in \Re^{2m}$ contains the current value of the f^{th} input variable. Finally, combining the expressions for all the variables in block i a condensed expression is obtained.

$$\begin{bmatrix} V_B & 0 & \cdot & 0 \\ 0 & V_B & \cdot & 0 \\ \cdot & \cdot & \ddots & \cdot \\ 0 & 0 & \cdot & V_B \end{bmatrix} \cdot \begin{bmatrix} \Delta u_{i_1} \\ \Delta u_{i_2} \\ \vdots \\ \Delta u_{i_{m_i}} \end{bmatrix} + \begin{bmatrix} \gamma_{i_1} \\ \gamma_{i_2} \\ \vdots \\ \gamma_{i_{m_i}} \end{bmatrix} \leq 0 \tag{3.24}$$

or

$$W_B \cdot \Delta u_i + \gamma_i \leq 0$$

Where $W_B \in \Re^{2m_i \cdot M_i \times m_i \cdot M_i}$ and $\gamma_i \in \Re^{2m_i \cdot M_i}$ contains the current values of the all the input variables in block i. Using the very same analysis on the input change constraints, $\Delta u_i^{\min} \leq \Delta u_i \leq \Delta u_i^{\max}$ can be expressed as follows.

$$\begin{bmatrix} I_B & 0 & \cdot & 0 \\ 0 & I_B & \cdot & 0 \\ \cdot & \cdot & \ddots & \cdot \\ 0 & 0 & \cdot & I_B \end{bmatrix} \cdot \begin{bmatrix} \Delta u_{i_1} \\ \Delta u_{i_2} \\ \vdots \\ \Delta u_{i_{m_i}} \end{bmatrix} + \begin{bmatrix} \delta_{i_1} \\ \delta_{i_2} \\ \vdots \\ \delta_{i_{m_i}} \end{bmatrix} \leq 0 \tag{3.25}$$

$$I_B = \begin{bmatrix} I \\ -I \end{bmatrix} \qquad \delta_{i_f} = \begin{bmatrix} -\nu \cdot \Delta u_{i_f}^{\max} \\ -\nu \cdot \Delta u_{i_f}^{\min} \end{bmatrix}$$

Which can be further condensed as follows.

$$II_B \cdot \Delta u_i + \delta_i \leq 0 \tag{3.26}$$

Where $I \in \Re^{m_i x m_i}$ is an identity matrix and $I_B \in \Re^{2m_i x m_i}$ is a stacked matrix composed of two identity matrices with opposite signs and $II_B \in \Re^{2m_i \cdot M_i x m_i \cdot M_i}$.

Finally the optimization problem (3.18) can be rewritten in the following way.

$$\begin{aligned}
&\underset{\Delta u_i}{Min}\ J_i = \frac{1}{2}\Delta u_i^T \left(A_{ii}^T Q_i A_{ii} + R_i\right)\Delta u_i + \left[A_{ii}^T Q_i \left(e_i + E_i\right)\right] \Delta u_i + \frac{1}{2}\left(e_i + E_i\right)^T Q_i \left(e_i + E_i\right) \\
&\textit{subject to} \\
&\qquad W_B \cdot \Delta u_i + \gamma_i \le 0 \\
&\qquad II_B \cdot \Delta u_i + \delta_i \le 0 \qquad\qquad (3.27) \\
&\qquad e_i = y_i^p + d_i - y_i^{sp} \\
&\qquad E_i = \sum_{\substack{j=1 \\ j \ne i}}^{N} A_{i,j} \cdot \Delta u_j
\end{aligned}$$

The Lagrangian of this problem is given in the following.

$$\begin{aligned}
L_i\left(\Delta u_i, \lambda_i, \phi_i\right) = &\frac{1}{2}\Delta u_i^T \left(A_{ii}^T Q_i A_{ii} + R_i\right)\Delta u_i + \left[A_{ii}^T Q_i \left(e_i + E_i\right)\right] \Delta u_i \\
&+ \frac{1}{2}\left(e_i + E_i\right)^T Q_i \left(e_i + E_i\right) \\
&+ \lambda_i^T \cdot \left(W_B \cdot \Delta u_i + \gamma_i\right) \\
&+ \phi_i^T \cdot \left(II_B \cdot \Delta u_i + \delta_i\right)
\end{aligned} \qquad (3.28)$$

Where $\lambda_i \in \Re^{2m_i \cdot M_i}$ and $\phi_i \in \Re^{2m_i \cdot M_i}$ are the Lagrange multipliers for the input and move size constraints respectively. The solution to this convex quadratic optimization problem must satisfy the following first-order KKT conditions (Biegler et al. 1997).

[A] $$\nabla_{\Delta u_i} L_i(\Delta u_i, \lambda_i, \phi_i) = \left(A_{ii}{}^T Q_i A_{ii} + R_i\right)\Delta u_i + A_{ii}{}^T Q_i (e_i + E_i) + W_B^T \cdot \lambda_i + II_B^T \cdot \phi_i = 0 \quad (3.29a)$$

[B] $$W_B \cdot \Delta u_i + \gamma_i \leq 0 \quad (3.29b)$$

$$II_B \cdot \Delta u_i + \delta_i \leq 0 \quad (3.29c)$$

[C] $$\lambda_{i_{ff}} \cdot \left(W_B \cdot \Delta u_i + \gamma_i\right)_{ff} = 0 \qquad ff = 1, \cdots, M_i \cdot m_i \quad (3.29d)$$

$$\phi_{i_{ff}} \cdot \left(II_B \cdot \Delta u_i + \delta_i\right)_{ff} = 0 \qquad ff = 1, \cdots, M_i \cdot m_i \quad (3.29e)$$

[D] $$\left(\lambda_i, \phi_i\right) \geq 0 \quad (3.29f)$$

The first-order KKT conditions have five different components that are necessary for optimality: [A] *Stationary or Linear dependence of gradients* [B] *Feasibility* [C] *Complementarity* [D] *Nonnegativity.* Finally [E] covers two separate issues that can also be considered. (a) A constraint qualification can be considered to account for degeneracy problems of the active constraints. Basically a strict local minimizer, Δu_i^* must also satisfy a *Strict complementarity* condition, which state that exactly one either the constraint or its associated Lagrange multiplier is zero but not both. (b) A second order condition to distinguish local minimizer from other stationary points requires the Hessian matrix at Δu_i^* to be positive definite on the null space of the active constraints.

[E (a)] *Strict complementarity*

and

[E (b)] $$Z^T \left(\nabla^2 J_i\left(\Delta u_i^*\right)\right) Z \text{ is positive definite}$$

Where Z is a null-space matrix for the matrix of active constraints at Δu_i^*. These conditions are only necessary and does not require that the Hessian of the Lagrangian itself be positive definite. It is a less stringent requirement. However if the Hessian of the Lagrangian at Δu_i^* is positive semidefinite, $\left(eig(\nabla^2_{\Delta u_i} L_i) \geq 0\right)$, which is the case for a well-posed MPC problem, then of course the second condition will be satisfied.

3.3.2 Simultaneous Solution for the Constrained D-MPC

The main idea behind the coordination method proposed in this work is to transform several optimization problems (MPC) into a single-level problem by replacing each problem with its optimality conditions, and then, simultaneously solving the resulting set of equations. Section 3.2 showed that for an unconstrained D-MPC only a set of linear equations is to be solved. The constrained case on the other hand becomes more challenging due to the set of complementarity conditions (3.29d) and (3.29e) that arise from the inequality constraints. These conditions introduce nonconvexity into the problem thus making the solution of the problem much more difficult. The simultaneous D-MPC problem is described in (3.30).

In order to demonstrate the simultaneous procedure for the constrained case let us consider the simplest case that involves two blocks ($N = 2$), each block with a single input single output controller ($M_1, P_1, M_2, P_2 = 1$) and a single input and output horizon ($m_1, p_1, m_2, p_2 = 1$).

	KKT 1	KKT 2
[A]	$\left(A_{11}^T Q_1 A_{11} + R_1\right)\Delta u_1 + \left(A_{11}^T Q_1 A_{12}\right)\Delta u_2 + A_{11}^T Q_1 (e_1)$ $+ W_B^T \cdot \lambda_1 + II_B^T \cdot \phi_1 = 0$	$\left(A_{22}^T Q_2 A_{22} + R_2\right)\Delta u_2 + \left(A_{22}^T Q_2 A_{21}\right)\Delta u_1 + A_{22}^T Q_2 (e_2)$ $+ W_B^T \cdot \lambda_2 + II_B^T \cdot \phi_2 = 0$
[B]	$W_B \cdot \Delta u_1 + \gamma_1 \leq 0$ $II_B \cdot \Delta u_1 + \delta_1 \leq 0$	$W_B \cdot \Delta u_2 + \gamma_2 \leq 0$ $II_B \cdot \Delta u_2 + \delta_2 \leq 0$
[C]	$\lambda_{1ff}\left(W_B \cdot \Delta u_1 + \gamma_1\right)_{ff} = 0 \quad ff = 1, \cdots, M_1 \cdot m$ $\phi_{1ff}\left(II_B \cdot \Delta u_1 + \delta_1\right)_{ff} = 0 \quad ff = 1, \cdots, M_1 \cdot m$	$\lambda_{2ff}\left(W_B \cdot \Delta u_2 + \gamma_2\right)_{ff} = 0 \quad ff = 1, \cdots, M_2 \cdot m$ $\phi_{2ff}\left(II_B \cdot \Delta u_2 + \delta_2\right)_{ff} = 0 \quad ff = 1, \cdots, M_1 \cdot m$
[D]	$(\lambda_1, \phi_1) \geq 0$	$(\lambda_2, \phi_2) \geq 0$

(3.30)

The geometrical interpretation of this problem can be visualized in Figure 3.2 for the two-dimension problem where the presence of regulatory constraints can modify the number and location of the equilibrium points.

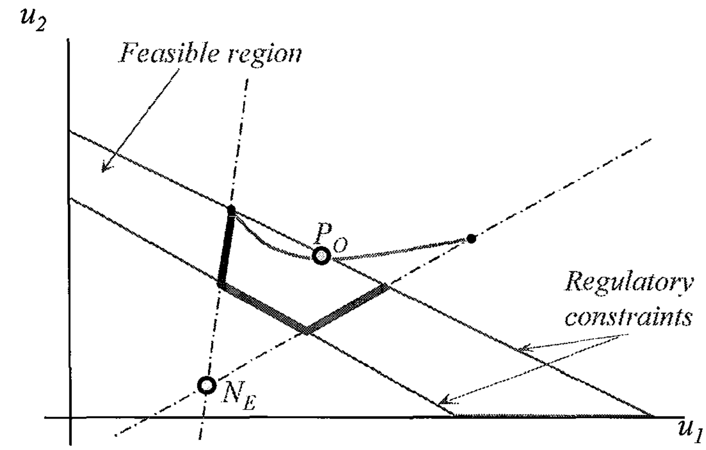


Figure 3.2 Intersection of KKT Conditions

The solution of the simultaneous D-MPC problem in (3.30) is challenging due to the complementarity conditions that introduce a combinatorial feature to the problem. This could be formulated as a mixed integer optimization problem, which would grow in size exponentially. Alternatively, one could formulate this problem with all continuous variables. Recently, several works (Raghunathan et al. 2003; Baker and Swartz, 2008) have reported successful solution of this class of problems using continuous variables by using an interior point solver called IPOPT-C. This solver has a specific modification in order to handle the complementarity equations. An important advantage in this methodology is that the simultaneous approach overcomes the necessity of an iterative algorithm. A disadvantage however is the non-convexity of the problem, which makes assurance of a global solution using IPOPT-C problematic.

Problem (3.31) has zero degrees of freedom for optimization. Therefore, in this work and in order to use the capabilities of the IPOPT-C solver a "false" objective function is incorporated. The resulting D-MPC problem is the following.

$$\min_{(\Delta u,\lambda,\phi)} F(\Delta u,\lambda,\phi) = c$$
$$\textit{subject to}$$
$$\left\{\begin{array}{c}[A]_i \\ [B]_i \\ [C]_i \\ [D]_i\end{array}\right\} \quad i = 1,\cdots,N \tag{3.31}$$

Where c is just a constant and $[A]_i$,$[B]_i$,$[C]_i$ and $[D]_i$ are the first-order KKT conditions of the controller in block i as described in (3.29).

The approach was successful in test cases for systems with positive Block Relative Gain. However, in systems with a negative BRG configuration (3.31) achieved solutions that were obviously incorrect based on engineering insight. For example in one of the Negative BRG simulations, offset was often obtained even when no input saturation needed to be active. The details on the case studies will be addressed in Chapter 5.

In order to overcome the complexity caused by the complementarity constraints an alternative approach was developed to ensure tractable real-time computation. This alternative approach basically consists in implementing a heuristic active set strategy that is able to remove non-convexity of the problem while achieving the autonomy sought for the D-MPC and retaining the coordination among blocks.

3.4 Heuristic Approach to Constrained D-MPC

The first attempt to remove the complementarity problem was to use the so-called DMC Heuristic, first presented by Prett and Gillete (1980). This method consists in incorporating input constraints as they are becoming active. However their method does not include the Lagrange multipliers for the active constraints in the stationary equations. This method has been proven to be effective in practice for single (centralized) MPC. However the Prett and Gillete DMC heuristic is not viable for the Decentralized MPC problem because not including Lagrange multipliers will cause the lost of local autonomy.

An alternative method is then proposed to solve the Decentralized MPC problem. This method is also based on an active set strategy, which has been used in MPC QP solvers. A similar method was developed for the Model Predictive Heuristic Controller developed by Richalet et al. (1987) for centralized MPC applications.

The first step in the development of the heuristic method is to define the way the constraints are addressed. Prett and Gillete, (1980) call them *time variant constraints* and described them as follows.

"*They are not always active. Depending on plant measurements and conditions they may or may not be activated. Hence, they become integral parts of the control system model only when they have been activated. At all other times, they are invisible to the control model. In case one manipulated variable hits high or low limits, in that case it can only move in one direction away from the limit.*"

These equations represent the set of constraints that have become active, and therefore must be enforced. Assume for the moment that the status of every inequality constraint is known, i.e. the active set is known.

The optimization problem for block i including the active constraint is the following.

$$\begin{aligned} &\underset{\Delta u_i}{Min}\ J_i = \frac{1}{2}\Delta u_i^T\left(A_{ii}^T Q_i A_{ii} + R_i\right)\Delta u_i + \left[A_{ii}^T Q_i\left(e_i + E_i\right)\right]\Delta u_i + \frac{1}{2}\left(e_i + E_i\right)^T Q_i\left(e_i + E_i\right) \\ &subject\ to \\ &\qquad H_i \cdot \Delta u_i - B_i = 0 \end{aligned} \tag{3.32}$$

Where the equality constraints represent only the known active constraints. The Lagrangian of this problem is.

$$\begin{aligned} L_i(\Delta u_i, \lambda_i) = &\frac{1}{2}\Delta u_i^T\left(A_{ii}^T Q_i A_{ii} + R_i\right)\Delta u_i + \left[A_{ii}^T Q_i\left(e_i + E_i\right)\right]\Delta u_i + \frac{1}{2}\left(e_i + E_i\right)^T Q\left(e_i + E_i\right) \\ &+ \lambda_i^T \cdot \left(H_i \cdot \Delta u_i - B_i\right) \end{aligned} \tag{3.33}$$

And the corresponding optimality conditions are the following.

[A]

$$\nabla_{\Delta u_i} L_i(\Delta u_i, \lambda_i) = \left(A_{ii}{}^T Q_i A_{ii} + R_i\right)\Delta u_i + \left[A_{ii}{}^T Q_i (e_i + E_i)\right] + H_i^T \cdot \lambda_i = 0 \qquad (3.34a)$$

$$\nabla_{\lambda_i} L_i(\Delta u_i, \lambda_i) = H_i \cdot \Delta u_i - B_i = 0 \qquad (3.34b)$$

This can be rearranged in the following linear system for block i.

$$\begin{bmatrix} A_{ii}{}^T Q_i A_{ii} + R_i & H_i{}^T \\ H_i & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta u_i \\ \lambda_i \end{bmatrix} = \begin{bmatrix} -A_{ii}{}^T Q_i (e_i + E_i) \\ B_i \end{bmatrix} \qquad (3.35)$$

For each active constraint an extra row is added to the linear system along with its respective Lagrange multiplier in the vector of variables.

For example consider having the following active constraint on the second time step $(k+2)$ of the input variable $(f = 1)$ in block i.

$$u_{i_{1,(k+2)}} = u_{i_{1,(k)}} + \Delta u_{i_{1,(k)}} + \Delta u_{i_{1,(k+1)}} = u_{i_1}^{\max}$$

This constraint can be extracted from the matrix form.

$$\begin{bmatrix} u_{i_{1,(k+1)}} \\ u_{i_{1,(k+2)}} \\ \vdots \\ \vdots \\ u_{i_{1,(k+m)}} \end{bmatrix} = \begin{bmatrix} u_{i_{1,(k)}} \\ u_{i_{1,(k)}} \\ \vdots \\ \vdots \\ u_{i_{1,(k)}} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta u_{i_{1,(k)}} \\ \Delta u_{i_{1,(k+1)}} \\ \vdots \\ \vdots \\ \Delta u_{i_{1,(k+m-1)}} \end{bmatrix} \qquad (3.36)$$

In this case where the calculated variable $u_{i_{1,(k+2)}}$ in block i is active, the corresponding row (2nd) is selected as a constraint and added to H_i.

$$H_i = \begin{bmatrix} 1 & 1 & 0 & \cdots & 0 \end{bmatrix} \qquad (3.37)$$

Repeating the same procedure for each block MPC controller $(i = 1,\cdots,N)$ to be coordinated results in a set of the stationary and feasibility conditions, which in turn is an augmented version of the unconstrained D-MPC controller presented in (3.16). The controller including the active constraints may be expressed as follows.

$$\begin{bmatrix} A_{DMPC} & H^T \\ H & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta u \\ \lambda \end{bmatrix} = \begin{bmatrix} b_{DMPC} \\ B \end{bmatrix} \tag{3.38}$$

$$A_{DMPC} = \left(A_D{}^T Q\, A_C + R\right) \qquad b_{DMPC} = A_D{}^T Q(-e)$$

Where λ is the Lagrange multiplier that appears due to the active constraints. The term B contains the collection of maximum changes for each input variable (i.e. $B_{i_l} = u_{i_l}^{\max} - u_{i_l,(k)}$), and H contains the collection of coefficients for the active constraints.

$$\lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \end{bmatrix} \qquad H = \begin{bmatrix} H_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & H_N \end{bmatrix} \qquad B = \begin{bmatrix} B_1 \\ \vdots \\ B_N \end{bmatrix} \tag{3.39}$$

The variables and parameters (λ_i, H_i, B_i) have the corresponding information for the active constraints in block i. Again, this is just a system of linear equations; where in order to have a solution matrix A_{DMPC} must be a full rank matrix.

Equations (3.38) provide the solution for a known active set. Now the basic logic of the complete algorithm that includes determining the active set is described.

Active Set Iterative Algorithm.

1. **Initialization:** Solve unconstrained optimization formulation.
2. **Find initial working set** of active constraints.
   2.1. Start Iterations.
   Pick the first active constraint along the input time horizon.
   Add the active constraint $Hu^{(j)} \cdot \Delta u = B_u^{(j)}$ to the unconstrained problem (3.16).
   Solve a resulting linear system in the form on (3.38)
   2.2. If there are constraint violations at 2.1.3 go to 2.1.1 if not to go to **3.**
3. **Done**

The heuristic algorithm provides a feasible active set for the constrained problem. The main difference in this strategy with a classic active set strategy is that once a constraint is fixed it never becomes inactive. The advantage is that a feasible solution is obtained within a finite number of iterations. A disadvantage is a possible non-optimal solution.

There is the possibility of several input violations at the same time. For such cases, the order in which an input variable (u) with violations is selected is based on the distance from its current value to the active bound; if two input variables have violations, the one closer to the active bound is selected to be set to its bound value.

It is important to mention that this strategy only considers hard input constraints. The addition of soft output constraints would require a different approach to handle the slack variables.

In conclusion, this very simple strategy only requires the solution of a linear set of equations at each iteration and a finite number of iterations. Computational experience comparing this method with a method solving the full non-linear KKT conditions is presented in the performance of the case studies in Chapter 5.

3.5 Comparison of D-MPC with Conventional Centralized MPC Controller

A common subject that often comes to mind is whether the D-MPC performance can be achieved by simply tuning a centralized MPC controller. In this section that question is addressed. First, the mathematical differences among controllers are presented with the objective to find if there is a way to achieve an equivalent dynamic performance, i.e., the same adjustments to the manipulated variables, using a centralized MPC formulation. Second, the role of the tuning parameters is explored and conclusions are drawn.

For the sake of simplicity, a *two by two* example without inequality constraints is used to compare both controllers, the analysis assumes perfect models. This case considers a controllable plant that includes dynamic interaction matrices $A_{1,2}$ and $A_{2,1}$ that are nonzero, i.e., process interaction exists. The corresponding set of optimality conditions for the D-MPC and the C-MPC are presented in (3.40) and (3.41) respectively. The centralized MPC formulation is described in detail in Appendix A.

D-MPC
$$\begin{bmatrix} A_{1,1}{}^T Q_{D1} A_{1,1} + R_{D1} & A_{1,1}{}^T Q_{D1} A_{1,2} \\ A_{2,2}{}^T Q_{D2} A_{2,1} & A_{2,2}{}^T Q_{D2} A_{2,2} + R_{D2} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} + \begin{bmatrix} A_{1,1}{}^T Q_{D1}(e_1) \\ A_{2,2}{}^T Q_{D2}(e_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.40}$$

C-MPC
$$\begin{bmatrix} A_{1,1}{}^T Q_{C1} A_{1,1} + A_{2,1}{}^T Q_{C2} A_{2,1} + R_{C1} & A_{1,1}{}^T Q_{C1} A_{1,2} + A_{2,1}{}^T Q_{C2} A_{2,2} \\ A_{1,2}{}^T Q_{C1} A_{1,1} + A_{2,2}{}^T Q_{C2} A_{2,1} & A_{1,2}{}^T Q_{C1} A_{1,2} + A_{2,2}{}^T Q_{C2} A_{2,2} + R_{C2} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} + \begin{bmatrix} A_{1,1}{}^T Q_{C1}(e_1) + A_{2,1}{}^T Q_{C2}(e_2) \\ A_{1,2}{}^T Q_{C1}(e_1) + A_{2,2}{}^T Q_{C2}(e_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.41}$$

We begin by comparing both equations element by element. It can be seen that every element of the C-MPC controller has an extra term. Now, the objective is to show that for a given set of D-MPC tuning parameters (Q_D, R_D), there is not a *simple* procedure to obtain a set of C-MPC tuning parameters (Q_C, R_C) that would drive both controllers to produce the same input action Δu with $e \neq 0$.

$$\left\lfloor A_{DMPC}^{-1} \cdot b_{DMPC} \right\rfloor = \Delta u = \left\lfloor A_{CMPC}^{-1} \cdot b_{CMPC} \right\rfloor \tag{3.42}$$

The term *simple* within the context of this analysis refers to obtaining tuning parameters by some direct algebraic procedure that does not involve a complex optimization problem. In this context the next discussion considers the most evident case in which both controllers could match each other. Such case would require every block element on both equations (3.40) and (3.41) to be exactly the same.

3.5.1 Tuning Parameters Effects.

The tuning parameters Q define the relative importance of the control of a specific output variable. In this way if Q_i is set to zero then outputs in block i are not controlled at all. For the case in question we will first consider the effect of this tuning parameter Q while setting the other tuning parameter R to zero. In order for the C-MPC in (3.41) to have the same performance as the D-MPC, the second term on each element of the first *row* must be zero. In the same way the first term of each element of the second *row* must be zero as shown below.

$$\begin{bmatrix} A_{1,1}^T Q_{C1} A_{1,1} + [\ 0\ ] + R_{C1} & A_{1,1}^T Q_{C1} A_{1,2} + [\ 0\ ] \\ [\ 0\ ] + A_{2,2}^T Q_{C2} A_{2,1} & [\ 0\ ] + A_{2,2}^T Q_{C2} A_{2,2} + R_{C2} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} + \begin{bmatrix} A_{1,1}^T Q_{C1}(e_1) + [\ 0\ ] \\ [\ 0\ ] + A_{2,2}^T Q_{C2}(e_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.43}$$

However, the only way to obtain this result would require setting the tuning parameters Q_{C1} and Q_{C2} equal to zero, which in turn would make the entire matrix equal to zero and therefore singular. From this observation it can be concluded that there is no way to match both controllers in an element-by-element fashion by means of changing parameter Q with a fixed parameter R.

The suppression factor, R on the other hand has the function of penalizing the magnitude of change of an input variable. From (3.40) and (3.41) it can be observed that R has no effect on the off-diagonal block elements of the matrix nor on the feedback information terms. Therefore, the suppression factor has no effect on the off-diagonal blocks of the controllers, and consequently, these off-diagonal terms would never be equal in both controllers.

However, it is important to mention that if a there were to exist a combination of tuning parameters (Q_C, R_C) and (Q_D, R_D) that makes equation (3.42) hold, obtaining such parameters would require the solution of complicated nonlinear problem with no guarantee of solution.

More importantly, the tuning would only be valid for one scenario. This would require the solution of a non-linear, non-convex tuning problem as part of every MPC controller execution. We deem this to be an unacceptable burden for real-time implementation.

In summary, by means of ***simple*** tuning the typical formulation of a centralized MPC is not able to reach autonomous decentralized goals just as the D-MPC controller is not able to reach centralized goals.

This section made clear that no *simple* tuning procedure and maybe no procedure at all exist in order to achieve autonomous goals using a centralized MPC controller.

3.6 Extended D-MPC Formulation

In this section an extended D-MPC controller is presented with the aim of achieving the range of solutions between D-MPC and C-MPC. The controller is formulated using the same method as the D-MPC controller described in Section 3.1. Basically, each local controller is formulated separately and then is replaced with its optimality conditions. The difference is that this time each local controller includes a term that accounts for the effect on other blocks' output control performance with its importance determined by a different weighting factor matrix $(W_{i,j})$. This tuning parameter $W_{i,j}$ indicates the weighting factors for outputs in block $j \neq i$ to be used in the objective function of block i. In this manner, the unconstrained MPC controller in block i has the following formulation. (Here, we will consider the controller without inequality constraints.)

$$\underset{\Delta u_i}{Min}\ J_i = \frac{1}{2} \cdot \left(\left\| y_i - y_i^{sp} \right\|_{Q_i}^2 + \sum_{\substack{j=1 \\ j \neq i}}^{N} \left\| y_j - y_j^{sp} \right\|_{W_{ij}}^2 + \left\| \Delta u_i \right\|_{R_i}^2 \right) \tag{3.44}$$

$$\begin{aligned} y_i &= A_{i,i} \Delta u_i + E_i + y_i^p + d_i \\ y_j &= A_{j,j} \Delta u_j + E_j + y_j^p + d_j \quad j \neq i \end{aligned} \tag{3.45}$$

$$E_i = \sum_{\substack{k=1 \\ k \neq i}}^{N} A_{i,k} \cdot \Delta u_k \qquad E_j = \sum_{\substack{k=1 \\ k \neq j}}^{N} A_{j,k} \cdot \Delta u_k \tag{3.46}$$

Where

$$\Delta u_i \in \Re^{(m_i \cdot M_i)} \qquad y_i, y_i^{sp}, y_i^p, d_i, \in \Re^{(p_i \cdot P_i)}$$

$$A_{i,i} \in \Re^{(p_i \cdot P_i) \times (m_i \cdot M_i)} \qquad y_j, y_j^{sp}, y_j^p, d_j, \in \Re^{(p_j \cdot P_j)}$$

$$Q_i \in \Re^{(p_i \cdot P_i) \times (p_i \cdot P_i)} \qquad A_{i,j} \in \Re^{(p_i \cdot P_i) \times (m_j \cdot M_j)} \quad j \neq i$$

$$R_i \in \Re^{(m_i \cdot M_i) \times (m_i \cdot M_i)} \qquad W_{ij} \in \Re^{(p_j \cdot P_j) \times (p_j \cdot P_j)}$$

$$N = Number\ of\ Blocks$$

Applying the same procedure described in Section 3.1 (i.e. evaluating the stationary conditions for the interacting controllers with coordinated variables and solving them simultaneously) for a two block example results in the following linear system of equations.

$$\begin{gathered} \begin{bmatrix} A_{1,1}{}^T Q_1 A_{1,1} + A_{2,1}{}^T W_{1,2} A_{2,1} + R_1 & A_{1,1}{}^T Q_1 A_{1,2} + A_{2,1}{}^T W_{1,2} A_{2,2} \\ A_{1,2}{}^T W_{2,1} A_{1,1} + A_{2,2}{}^T Q_2 A_{2,1} & A_{1,2}{}^T W_{2,1} A_{1,2} + A_{2,2}{}^T Q_2 A_{2,2} + R_2 \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} \\ + \begin{bmatrix} A_{1,1}{}^T Q_1 (e_1) + A_{2,1}{}^T W_{1,2} (e_2) \\ A_{1,2}{}^T W_{2,1} (e_1) + A_{2,2}{}^T Q_2 (e_2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{gathered} \tag{3.47}$$

The similarities of this Extended D-MPC and the C-MPC controller in (3.41) are evident. The resulting controller can be separated as follows.

$$\begin{bmatrix} A_{1,1}{}^T Q_1 A_{1,1} + R_1 & A_{1,1}{}^T Q_1 A_{1,2} \\ A_{2,2}{}^T Q_2 A_{2,1} & A_{2,2}{}^T Q_2 A_{2,2} + R_2 \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} + \begin{bmatrix} A_{1,1}{}^T Q_1 (e_1) \\ A_{2,2}{}^T Q_2 (e_2) \end{bmatrix}$$
$$+ \begin{bmatrix} A_{2,1}{}^T W_{1,2} A_{2,1} & A_{2,1}{}^T W_{1,2} A_{2,2} \\ A_{1,2}{}^T W_{2,1} A_{1,1} & A_{1,2}{}^T W_{2,1} A_{1,2} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix} + \begin{bmatrix} A_{2,1}{}^T W_{1,2} (e_2) \\ A_{1,2}{}^T W_{2,1} (e_1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.48)$$

Then it can be easily redefined in the following way

$$\left[\left(A_D{}^T Q \cdot A_C + R \right) \Delta u + A_D{}^T Q \cdot (e) \right] + \left[\left(A_{OD}{}^T W \cdot \Pi\, A_C \right) \Delta u + A_{OD}{}^T W \cdot \Pi (e) \right] = 0 \quad (3.49)$$

Where the following matrices can be defined.

$$W = \begin{bmatrix} 0 & W_{1,2} & \cdots & W_{1,N} \\ W_{2,1} & 0 & \cdots & \vdots \\ \vdots & \cdots & \ddots & W_{N-1,N} \\ W_{N,1} & \cdots & W_{N,N-1} & 0 \end{bmatrix}^T \text{ and } A_{OD} = \begin{bmatrix} 0 & A_{1,2} & \cdots & A_{1,N} \\ A_{2,1} & 0 & \cdots & A_{2,N} \\ \cdots & \cdots & \ddots & \cdots \\ A_{N,1} & A_{N,2} & \cdots & 0 \end{bmatrix} \quad (3.50)$$

And Π is a permutation matrix that switches the block-rows of matrix A_C. Finally, the control actions are calculated as follows.

$$\Delta u = \left[\underbrace{A_D{}^T Q \cdot A_C + R}_{I} + \underbrace{A_{OD}{}^T W \cdot (\Pi\, A_C)}_{II} \right]^{-1} \cdot \left[\underbrace{A_D{}^T Q}_{I} + \underbrace{A_{OD}{}^T W \cdot \Pi}_{II} \right] \cdot (-e) \quad (3.51)$$

The terms underlined with I refer to the D-MPC formulation and the terms with II are the additional terms required to achieve the C-MPC performance. The following observations can be made.

- As $W_{2,1}$ and $W_{1,2}$ approach zero the performance approaches that of the D-MPC controller.
- As $W_{2,1}$ and $W_{1,2}$ approach Q_1 and Q_2 respectively a C-MPC controller is approached. This can be easily visualized by comparing equations (3.47) with (3.41).

Figure 3.3 presents a sketch of the range of performance for the Extended D-MPC controller as the parameter W is modified. For example, if parameter W is set equal to a certain percentage of parameter Q then the solution will fall between the points P_0 (the original C-MPC solution) and N_E (the D-MPC solution) depicted in the figure.

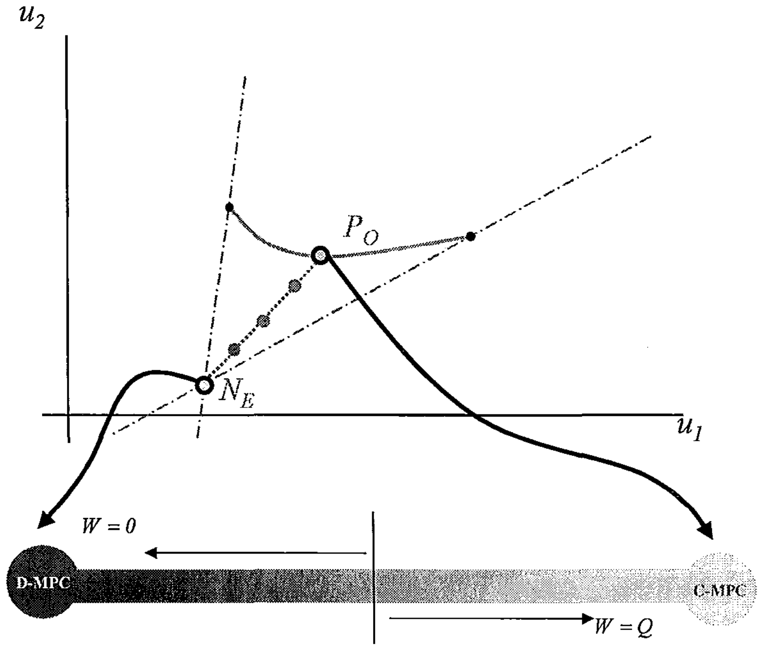


Figure 3.3 Effect of Tuning Parameter W

It is important to mention that the case where $W = Q$ results in the Cooperative MPC controller developed by Venkat and Rawlings (2004), which from its conception has a different design objective. This objective is to have distributed computation in a decentralized MPC that has performance close to the centralized controller, C-MPC. Clearly, this is a different goal, centralized versus local autonomy; therefore, we will not apply the Venkat and Rawlings controller.

A concluding remark regarding the Extended D-MPC presented in this section is that the spectrum of possibilities that ranged form D-MPC to C-MPC can be readily analyzed by means of a simple parameter $(W_{i,j})$, which is also useful to understand and define the goals of the coordinated controller.

3.7 Summary and Conclusion

The D-MPC controller is developed parting from a strategy similar to that of multilevel optimization. Here several optimization problems at a same level are replaced with their respective optimality conditions and then solved simultaneously. It can be concluded that applying this strategy to a set of unconstrained MPC controllers will result in a D-MPC controller with a well-defined structure that is easier to visualize and analyze. Additionally the way tuning is addressed remains the same as in conventional MPC control.

Another important concluding remark is that there is not a practical way to achieve an autonomous D-MPC performance by means of tuning a conventional centralized MPC controller. In the same way it is impossible for the D-MPC controller to achieve a C-MPC performance.

Finally, implementing constrained control required more work. The strategy that includes KKT conditions is correct, however its implementation was sometimes unsuccessful in cases with negative BRG configurations. Although tuning and solver adjustments may help it was decided to try to remove the non-convexity by implementing a heuristic strategy to enhance the D-MPC formulation. The improved strategy consists in a systematic method that detects constraints violations and automatically incorporates required bounds (active set) into the controller formulation. The approach is computationally tractable yielding a small set of convex problems to be solved sequentially.

Chapter 4

Controllability and Stability Analysis

In this chapter, controllability and stability analysis of the D-MPC controller structure is presented. Controllability provides basic criteria that must be satisfied by the process for application of the D-MPC; it does not provide insight regarding the quality of control performance, but only a guarantee that feedback control is in some sense possible. Once a process has been deemed suitable for the application of the D-MPC controller, the next step is to analyze if the controller can provide closed loop stability. Classical stability criteria for discrete control systems are applied to the D-MPC controller formulated in state space. Different control structures such as multiple SISO controllers or sets of multivariable blocks can be easily tested for nominal stability. As a result, this chapter provides methods for selecting process applications that could be suitable for D-MPC. Control performance will be evaluated in the next chapter.

4.1 Definitions of the Plants to be Controlled by the D-MPC.

We begin the analysis by refining the objectives of the implementation of a Block D-MPC that were presented in Chapter 1. By using the D-MPC controller we want to apply D-MPC to processes that can be controlled by a centralized MPC (C-MPC) controller.

Therefore, we consider any process for which C-MPC can be applied as a candidate for D-MPC, which provides autonomy for each block controller. In this way an implementation of the D-MPC controller must be able to:

1. Achieve output controllability, not full state controllability.
2. Maintain the steady-state output variable equal to the set point for changes in disturbances and set points by adjusting input variables u for unconstrained applications.
3. Achieve a good dynamic performance.

Two important characteristics must be analyzed in order to define the applicability of the D-MPC controller. First, the *controllability*, which is a characteristic of the process and second the *existence* of the controller solution.

4.1.1 Classical Controllability Definitions.

Several definitions of controllability can be found in literature, with the proper choice depending on the control application. We start with a simple definition of **steady-state controllability**, which is independent of the control algorithm (Marlin, 2000).

A system is controllable if the controlled variables can be maintained at their set points, in the steady-state, in spite of disturbances entering the system.

In this way a square process is **controllable** if the determinant of the steady-state gain matrix is nonzero. Moreover, square systems are not controllable if any of the following conditions occurs:

1. Any process inputs are linearly dependent (giving dependent columns)
2. Any process outputs are linearly dependent (giving dependent rows)
3. A process input does not influence any output (giving rows of zeros).

4. A process output is not influenced by any input (giving column of zeros)

In the same way a non-square system is controllable as long as the (column) rank of the gain matrix is equal to number of controlled variables.

Table 4.1 describes two other important definitions (Skogestad and Postlethwaite, 1996). However, it is important to note that these common controllability criteria are too restrictive for many process control applications.

Table 4.1 Controllability Definitions

	Definition	Shortcoming	Potential application
1. Pointwise State (or Output) Controllability	Indicates if an input variable is able to bring the states from any initial value to any final value within some time window	It does not imply the states (outputs) can be maintained at the "final conditions" at steady state.	Batch control
2. Functional Controllability	A system is functional controllable if given any *suitable* [1] output sequence there exists an input sequence which generates the output sequence.	The term suitable is too restrictive and therefore can't be applied to systems with RHP zeros.	Continuous processes where the entire defined trajectory must be achieved without error

In this work, only the simple, steady-state controllability criterion will be applied. The other dynamic criteria are deemed too restrictive. However, steady-state controllability may not provide sufficient insight to applications of D-MPC. Therefore, we proceed to further analysis that includes information about the controller as well as the process, so that it is not precisely controllability analysis.

4.1.2 Existence of a Centralized MPC Control Solution

[1] A suitable sequence is one, which does not ask for nonzero output in less time that the inherent time delay of the system, and which also has a z-transform (Rosenbrock, 1974).

As mentioned above the application of D-MPC proposed in this work is going to be limited to processes that have been deemed suitable for a C-MPC controller implementation. In this way the analysis in this section is based on the C-MPC controller. The C-MPC controller requires the inversion of a matrix in order to calculate the control law. If that matrix cannot be inverted then the control action cannot be implemented. The unconstrained C-MPC controller is described in Appendix A, and the control law for the unconstrained case can be expressed by the following equation.

$$\Delta u = \left(A_C{}^T Q\ A_C + R\right)^{-1} A_C{}^T Q\ \left(-e\right) \tag{4.1}$$

$$A_{CMPC} = \left(A_C{}^T Q\ A_C + R\right) \tag{4.2}$$

Where parameter Q is the output weighting matrix and has the form of a diagonal positive definite matrix. The matrix R is the move suppression factor and is a positive semidefinite matrix. In order for the centralized MPC controller to exist matrix A_{CMPC} as defined in (4.2) must be non-singular and therefore invertible for any suitable value of the tuning parameters R and Q which are diagonal matrices with positive (or non-negative) coefficients on their diagonals.

When determining whether A_{CMPC} is non-singular, a following useful matrix properties for positive definite matrices will be applied (Horn and Johnson, 1976).

P.1. The sum of any Positive Definite (*PD*) matrices of the same size is positive definite. More generally, any nonnegative linear combination of positive semidefinite matrices is also positive semidefinite.

P.2. Let $\Sigma \in M_n$ (Matrix of $n x n$ dimension) be Positive Definite. If $C \in M_{n,m}$ (Matrix of $n x m$ dimension) then $C^T \Sigma \cdot C$ is positive definite if and only if C has rank m.

Based on property P.1 and on the fact that parameter R is at least a positive semidefinite matrix the singularity analysis of expression (4.2) can be reduced to the following expression.

$$\hat{A}_{CMPC} = \left(A_C{}^T Q\ A_C\right) \tag{4.3}$$

(We note that even a process with no causal relationships between the manipulated and controlled variables can have a non-singular controller matrix by setting the move suppression elements to positive values. We would deem this situation non-controllable, as the solution would simply be a minimum effort controller, with all changes to the manipulated variables being zero.)

Furthermore, by applying property P.2 it can be guaranteed that as long as the dynamic matrix, A_C has a rank equal to its number of columns then the matrix $\left(A_C{}^T Q\ A_C\right)$ is positive definite and therefore non-singular. In this way the condition for invertibility of A_{CMPC} is the following:

$$Rank\,(A_C) = m \cdot M \tag{4.4}$$

Where M is the total number of input (manipulated) variables and m is the controller input horizon.

In summary, as long as the Q matrix is positive definite, which is the case for C-MPC controllers, the dynamic matrix A_C alone defines the applicability of the C-MPC controller. It is important to bear in mind that this matrix, A_C is built from step weights models that relate the dynamic effect of a specific input variable to a specific output variable.

4.1.3 Minimum Prediction Horizon

The objective of this section is to define the minimum prediction horizon necessary to ensure the full rank of A_C. In this way the first requirement is to use an output horizon, p, sufficiently large to reach the steady-state of the all the outputs in the dynamic model. Equivalently, the prediction horizon must be such that the last column of each dynamic matrix reaches the steady-state (or within some arbitrary small deviation from steady state). We propose that the proper prediction horizon can be calculated in the following way.

$$p = SS_{\max} + m - 1 \tag{4.5}$$

Where $SS_{\max}$ is the number of samples required to reach the steady-state of the slowest input-output process. This will generate a dynamic matrix as described in (4.6), which in turn guarantees a full column rank even for pure dead time processes.

$$A_{C g,f} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & \vdots \\ \vdots & a_2 & \ddots & 0 \\ a_{SS} & \cdot & \ddots & a_1 \\ a_{SS} & a_{SS} & \cdot & a_2 \\ \cdot & \cdot & \ddots & \vdots \\ a_{SS} & a_{SS} & \cdot & a_{SS} \end{bmatrix} \tag{4.6}$$

Where a_{SS} is the corresponding steady-state gain. A simple two by two example with pure dead time (τ_d) dynamics exemplifies the importance of this minimum horizon. The steady-state gain matrix and the MPC design for this hypothetical example is the following, with the gains from a distillation tower problem that will be considered throughout the thesis.

$$Kp = \begin{pmatrix} 0.07474 & -0.0667 \\ 0.1173 & -0.1253 \end{pmatrix} \quad \tau = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \quad \tau_d = \begin{pmatrix} 200 & 200 \\ 200 & 200 \end{pmatrix}$$

Input horizon	$m = 5$
Output horizon	$p = 5$
Sampling time	$\Delta t = 50$

Basically, the dead time is equal to four sampling times and then an input horizon $m = 5$ is selected. The output horizon is chosen to reach steady-state, in this case $p = 5$. For this selection the multivariable dynamic matrix will have the following form.

$$A_C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0747 & 0 & 0 & 0 & 0 & -0.0667 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1173 & 0 & 0 & 0 & 0 & -0.1253 & 0 & 0 & 0 & 0 \end{pmatrix}$$

This matrix is not full (column) rank, which in turn leads to a rank deficient $\hat{A}_{CMPC}$ matrix. This situation combined with a zero suppression factor $(R = 0)$ will produce a singular C-MPC controller. Evidently, choosing such tuning combination is to be avoided at all times. However, the intention in this section is to demonstrate the conditions that could lead to a non-singular matrix, A_C. The simplest solution is to modify the controller design by selecting the prediction horizon as defined in (4.5)

The matrix has now the following form.

$$A_C = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0747 & 0 & 0 & 0 & 0 & -0.0667 & 0 & 0 & 0 & 0 \\
0.0747 & 0.0747 & 0 & 0 & 0 & -0.0667 & -0.0667 & 0 & 0 & 0 \\
0.0747 & 0.0747 & 0.0747 & 0 & 0 & -0.0667 & -0.0667 & -0.0667 & 0 & 0 \\
0.0747 & 0.0747 & 0.0747 & 0.0747 & 0 & -0.0667 & -0.0667 & -0.0667 & -0.0667 & 0 \\
0.0747 & 0.0747 & 0.0747 & 0.0747 & 0.0747 & -0.0667 & -0.0667 & -0.0667 & -0.0667 & -0.0667 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.1173 & 0 & 0 & 0 & 0 & -0.1253 & 0 & 0 & 0 & 0 \\
0.1173 & 0.1173 & 0 & 0 & 0 & -0.1253 & -0.1253 & 0 & 0 & 0 \\
0.1173 & 0.1173 & 0.1173 & 0 & 0 & -0.1253 & -0.1253 & -0.1253 & 0 & 0 \\
0.1173 & 0.1173 & 0.1173 & 0.1173 & 0 & -0.1253 & -0.1253 & -0.1253 & -0.1253 & 0 \\
0.1173 & 0.1173 & 0.1173 & 0.1173 & 0.1173 & -0.1253 & -0.1253 & -0.1253 & -0.1253 & -0.1253
\end{pmatrix}$$

Taking the 10 rows containing the steady-state gain information forms a submatrix, A_{SX}, which can be easily visualized to be full column rank.

$$A_{SX} = \begin{pmatrix}
0.0747 & 0 & 0 & 0 & 0 & -0.0667 & 0 & 0 & 0 & 0 \\
0.0747 & 0.0747 & 0 & 0 & 0 & -0.0667 & -0.0667 & 0 & 0 & 0 \\
0.0747 & 0.0747 & 0.0747 & 0 & 0 & -0.0667 & -0.0667 & -0.0667 & 0 & 0 \\
0.0747 & 0.0747 & 0.0747 & 0.0747 & 0 & -0.0667 & -0.0667 & -0.0667 & -0.0667 & 0 \\
0.0747 & 0.0747 & 0.0747 & 0.0747 & 0.0747 & -0.0667 & -0.0667 & -0.0667 & -0.0667 & -0.0667 \\
0.1173 & 0 & 0 & 0 & 0 & -0.1253 & 0 & 0 & 0 & 0 \\
0.1173 & 0.1173 & 0 & 0 & 0 & -0.1253 & -0.1253 & 0 & 0 & 0 \\
0.1173 & 0.1173 & 0.1173 & 0 & 0 & -0.1253 & -0.1253 & -0.1253 & 0 & 0 \\
0.1173 & 0.1173 & 0.1173 & 0.1173 & 0 & -0.1253 & -0.1253 & -0.1253 & -0.1253 & 0 \\
0.1173 & 0.1173 & 0.1173 & 0.1173 & 0.1173 & -0.1253 & -0.1253 & -0.1253 & -0.1253 & -0.1253
\end{pmatrix}$$

A minimum singular value $(\sigma_{\min} = 0.004)$ of A_{SX}, proves that the sub matrix is non-singular. Based on the matrix property P.2 described above this design in turn guarantees matrix $\left(A_C{}^T Q \cdot A_C\right)$ to be positive definite and therefore a non-singular C-MPC controller.

4.1.4 Summary of Applicability Requirements.

In summary, in Section 4.1.1 was defined that as long as the Q matrix is positive definite, which is the case in C-MPC controllers, the dynamic matrix, A_C alone defines the applicability of the C-MPC controller.

Based on the dynamics of the process and on the existence of the controller the following is required in order to implement a C-MPC controller. Note that the first three points come from the "controllability" of the process and are necessary to guarantee the existence of the C-MPC controller.

1. Square systems or systems with more inputs than outputs.

2. Stable processes due to its step weight formulations.

3. For square systems the inverse of the steady-state gain matrix $(Kp)^{-1}$ must exists. For nonsquare systems the rank of the steady-state gain must be equal to the number of output variables. $rank(Kp) = P$

4. Use a sufficiently large output horizon, $p = SS_{\max} + m - 1$, which reaches the steady-state on the last column of every SISO dynamic matrix.

The fourth requirement on the other hand it is just sufficient in order to guarantee the existence of the C-MPC controller. As stated above the implementation of a D-MPC controller will be restricted to systems that can be controlled by a C-MPC controller.

4.2 Stability Analysis

This section focuses on analysing the nominal stability of the D-MPC control system. This analysis applies the classical linear stability analysis for discrete systems. The technical details are described along with a couple of numerical examples.

Inappropriate tuning parameters can yield an unstable D-MPC control system, even without model mismatch. Therefore, the selection of tuning parameters (Q, R, p) is restricted to the conditions described in the previous section. The stability analysis discussed in this section involves the application of a state space formulation. Additionally, when these results demonstrate nominal stability for a selected tuning they also provide a certificate for the existence (non-singularity) of the unconstrained D-MPC controller.

4.2.1 State Space Representation for C-MPC Using Step Response Models

The nominal stability of the coordinated D-MPC follows the methodology first proposed by Lee et al. (1994) for C-MPC controllers. This subsection briefly describes nominal stability analysis for C-MPC controllers, then in the next subsection the method is extended to D-MPC. Basically, the step weight model is transformed into a state space approximation, then the closed loop expression using an MPC controller is formed and the poles are analyzed. Thus the resulting state space model has p states where p is the prediction horizon for the controller. Finally it is important to note that in this work only nominal stability, which considers a perfect model, is considered.

First, let a_ℓ, $\ell = 1,2,\cdots,p$ denote the step response coefficients of a stable process. Then, the step response model A_S is defined as.

$$A_S = \begin{bmatrix} a_1 & a_2 & \cdots & a_p \end{bmatrix}^T \tag{4.7}$$

Now we define a recursive relationship for estimating the current and future value of the process output using step response models. Assume that at some point in time $(k-1)$ the p elements of the **state vector** $X(k-1)$ are known. (Note that the states are represented by captial $X(k)$, while the inputs and outputs are represented by the lower case Δu and y respectively. For example, $X(k-1)$ may be an initial steady state. Now assume that a change in the input variable at time $(k-1)$ is made, $\Delta u(k-1)$. The predictions $X(k)$ can be estimated using the previous predictions, $X(k-1)$ as follows.

$$\begin{aligned} X_1(k) &= X_2(k-1) + a_1 \Delta u(k-1) \\ X_2(k) &= X_3(k-1) + a_2 \Delta u(k-1) \\ \cdots & \quad \cdots \end{aligned} \tag{4.8}$$

Extending to the entire trajectory, $\ell = 1,2,\cdots,p-1$ gives.

$$X_\ell(k) = X_{\ell+1}(k-1) + a_\ell \Delta u(k-1) \tag{4.9}$$

Where ℓ refers to a time step in the future prediction performed at time ξ $(i.e.\ \ \xi = k, k-1)$ and k refers to a time step in the state space model.

The effect of $\Delta u(k-1)$ on the states $X_\ell(k)$ for $(\ell \geq p)$ is constant. Therefore, the effect for a unit step gives $a_\ell = a_p$ for $\ell \geq p$.

$$\begin{gathered} X_p(k) = X_p(k-1) \\ or \\ X_p(k) = X_{p-1}(k) \end{gathered} \tag{4.10}$$

It is important to note that for the purposes of this work, p is greater or equal to the minimum prediction horizon as defined in Section 4.1.3.

Equations (4.9) and (4.10) can be put in the following compact notation:

$$\begin{gathered} X(k) = F_0 \cdot X(k-1) + A_S \cdot \Delta u(k-1) \\ X(k) = [y_0(k) \quad y_1(k) \quad \cdots \quad y_{p-1}(k)]^T \end{gathered} \tag{4.11}$$

Remark: Each element of the state vector $X(k)$, $y_\ell(k)$ has the following interpretation: it denotes the next output value y at time $k+\ell$ assuming the input and disturbance remain constant starting at time $k-1$.

Then $\Delta u(k-1)$ is the manipulated variable change at time $k-1$ and $F_0 \in \Re^{p\times p}$ is a matrix of the following form.

$$F_0 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & 1 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \tag{4.12}$$

In this way the next output value is.

$$\begin{aligned} y(k) &= [1 \quad 0 \quad \cdots \quad 0]\cdot X(k) \\ y(k) &= N_S \cdot X(k) \end{aligned} \tag{4.13}$$

The next step is to calculate the control law for an MPC controller. The controller action at time k is computed based on the predicted state values. The future state values, $X(k+1)$ depend on the past predicted state values, $X(k)$ the current and future disturbance, Δd and the future manipulated variable changes, Δu. Then, the state values over a prediction p are defined as follows.

$$X(k+1) = F_0 \cdot X(k) + A \cdot \Delta u + A^d \cdot \Delta d \tag{4.14}$$

Where $A^d \in \Re^{px1}$ may contain the step response coefficients generated from the disturbance model or a vector of all ones when considering only a constant disturbance estimation, Δd is the disturbance entering the system at instant k, and the vector $\Delta u \in \Re^m$ is the manipulated variable change vector, consisting of controller moves in the control horizon m.

$$\Delta u = [\Delta u_{(k)} \quad \Delta u_{(k+1)} \quad \cdots \quad \Delta u_{(k+m-1)}]^T \tag{4.15}$$

Finally, $A \in \Re^{pxm}$ is the dynamic matrix as described in Section 3.2.

The controller action is computed by minimizing a quadratic norm of the difference between the predicted output and the set point.

$$\underset{\Delta u}{Min}\ J = \frac{1}{2}\cdot\left(\left\|X(k+1) - X_{SP}(k+1)\right\|_Q^2 + \left\|\Delta u\right\|_R^2\right) \tag{4.16}$$

Where $X_{SP}(k+1) = [y_{sp}(k+1) \ \cdots \ y_{sp}(k+p)]^T$ is the future output reference vector.

Remark: Due to the form in which the state vector $X(k+1)$ is defined it also corresponds to the minimization of the predicted output feedback error that can also be expressed as follows.

$$\underset{\Delta u}{Min}\ J = \frac{1}{2}\cdot\left(Q\sum_{\ell=1}^{p}\left(y(k+\ell) - y_{sp}(k+\ell)\right)^2 + R\sum_{\ell=0}^{m-1}\Delta u_{(k+\ell)}^{2}\right)$$

In this way the solution of the MPC controller using the state-space representation of a step model results in an expression of the following form (Lee et al, 1994).

$$\begin{aligned} \Delta u &= K_{MPC}\cdot\left(-e(k)\right) \\ K_{MPC} &= \left(A^T Q\ A + R\right)^{-1} A^T Q \\ e(k) &= F_0\cdot X(k) + A^d\cdot \Delta d(k) - X_{SP}(k) \end{aligned} \tag{4.17}$$

Where $K_{MPC} \in \Re^{mxp}$ is the MPC controller matrix that results from solving a least squares optimization problem and $e(k) \in \Re^{p}$ is the feedback vector entering the controller at each execution time. In general, this term $e(k)$ is composed of disturbance effects as well as the difference between the predicted output based on past-implemented changes in u and the set point.

Finally a closed loop dynamic equation for stability analysis is obtained. The input to this dynamic system is $X_{SP}(k-1)$ and the output is $y(k)$. The dynamic system is defined by the following set of equations that results from combining equations (4.14) and (4.17).

$$\begin{aligned} X(k) &= \hat{A} \cdot X(k-1) + B \cdot X_{sp}(k-1) \\ y(k) &= C \cdot X(k) \end{aligned} \tag{4.18}$$

Where $C = N_S$ and.

$$\hat{A} = [F_0 - A_S \cdot (K_{MPC}) \cdot F_0]$$

$$B = [A_S \cdot K_{MPC}]$$

The stability of the closed loop system is then determined by the eigenvalues of matrix $\hat{A}$ (Strang, 1980). The closed loop equation is stable whenever all the eigenvalues lie strictly within the unit circle (Lee et al, 1994). Naturally, the eigenvalues depend upon the tuning parameters.

4.2.2 Nominal Stability of Block D-MPC

The structure of the D-MPC controller allows for a straightforward application of the linear stability analysis described in detail above. An important characteristic of this stability analysis is that the block structure of the controller can be easily modified without affecting the structure of closed loop expression.

In this way the closed loop expression is maintained and only one term $(L \cdot K_{DMPC})$ is modified depending on the control structure. For example multiple SISO controllers, multivariable controllers or a combination of both can be easily evaluated for stability. An illustration is presented in Figure 4.1.

$$H_{SS} = F - \hat{A}_S \cdot \underbrace{(L \cdot K_{DMPC})} \cdot F$$

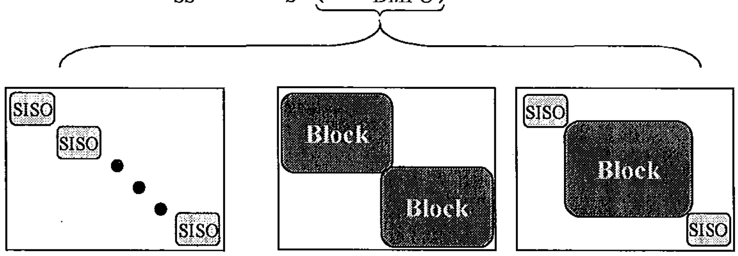


Figure 4.1 Stability expression for Different Control Structure

The unconstrained block D-MPC as described in Section 3.1 has a very similar structure to that of the Centralized MPC controller and can be expressed as follows:

$$\Delta u = K_{DMPC} \cdot (-e(k)) \tag{4.19}$$

Where

$$K_{DMPC} = \left(A_D{}^T Q\, A_C + R\right)^{-1} A_D{}^T Q$$

Then, based on (4.18) the closed loop equation for a **multivariable system** with multiple blocks using a D-MPC controller is the following.

$$\begin{aligned} X(k) &= \left[F - \hat{A}_S \cdot (L \cdot K_{DMPC}) \cdot F\right] \cdot X(k-1) + \left[\hat{A}_S \cdot K_{DMPC}\right] \cdot X_{sp}(k-1) \\ y_{(k)} &= C \cdot X(k) \end{aligned} \tag{4.20}$$

Hence the closed loop system is stable if and only if all the eigenvalues of the following expression H_{SS} lie strictly inside the unit circle.

$$H_{SS} = \left[F - \hat{A}_S \cdot (L \cdot K_{DMPC}) \cdot F\right] \tag{4.21}$$

Where F is now a square matrix with dimensions equal to of the total number output variables times the prediction horizon.

$$F = \begin{pmatrix} F_0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & F_0 & 0 \\ 0 & \cdots & 0 & F_0 \end{pmatrix} \in \Re^{p \cdot P \times p \cdot P} \tag{4.22}$$

With F_0 as defined in (4.12) and $\hat{A}_S$ now containing the step response models $A_{S_{g,f}}$ arranged in the following way.

$$\hat{A}_S = \begin{pmatrix} A_{S_{1,1}} & \cdots & A_{S_{1,M}} \\ \vdots & \ddots & \vdots \\ A_{S_{P,1}} & \cdots & A_{S_{P,M}} \end{pmatrix}$$

and

$$L = \begin{pmatrix} L_0 & O & O \\ O & \ddots & O \\ O & O & L_0 \end{pmatrix} \tag{4.23}$$

$$L_0 \in \Re^{1 \times m} \quad L_0 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$
$$O \in \Re^{1 \times m} \quad O = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}$$

Therefore, the closed loop stability of the multivariable system using a D-MPC controller is a function of the tuning parameters (Q, R). Again it is important to note that the structure of the vectors is properly handled by stacking multiple variables.

A couple of illustrative cases are now presented.

4.2.3 Stability of D-MPC – Numerical Cases

Case 1 – Single-variable Blocks: In order to show the capabilities of the D-MPC stability analysis consider the distillation column (Ogunnaike and Ray, 1994) where tray temperatures act as inferential variables for composition control. The outputs T_{21}, T_7 are the temperatures of trays 21 and 7, respectively and the inputs F_R, F_V denote the reflux flowrate and the vapor boilup flowrate to the distillation column. The nominal plant model is shown in (4.24).

$$\begin{pmatrix} T_{21} \\ T_7 \end{pmatrix} = \begin{pmatrix} \dfrac{32.63}{(99.6s+1)(0.35s+1)} & \dfrac{-33.89}{(98.02s+1)(0.42s+1)} \\ \dfrac{34.84}{(110.5s+1)(0.03s+1)} & \dfrac{-18.85}{(75.43s+1)(0.3s+1)} \end{pmatrix} \begin{pmatrix} F_V \\ F_R \end{pmatrix} \tag{4.24}$$

It is worth mentioning that Venkat and Rawlings (2004) used this case to prove that no communication MPC approach is able to stabilize a system with a negative RGA configuration. This system is then intentionally paired using $(F_R - T_{21})$ and $(F_V - T_7)$, resulting in the following RGA matrix.

$$\Lambda = \begin{pmatrix} -1.087 & 2.087 \\ 2.087 & -1.087 \end{pmatrix} \tag{4.25}$$

Note that the pairing gives poor integrity and violates the common convention of pairing on positive relative gains and BRG determinant.

For each MPC controller, an execution time of 10 sec is used, and the input and output horizon are $m = 25$ and $p = 125$ respectively. The tuning parameters used for this case are $Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $R = \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$. Figure 4.2 depicts the unit circle analysis using the stability expression in (4.21). It can be observed that D-MPC poles lie strictly inside the unit circle, which is not the case for the independent block (fully decentralized) MPC controller which has a couple of poles outside the unit circle.

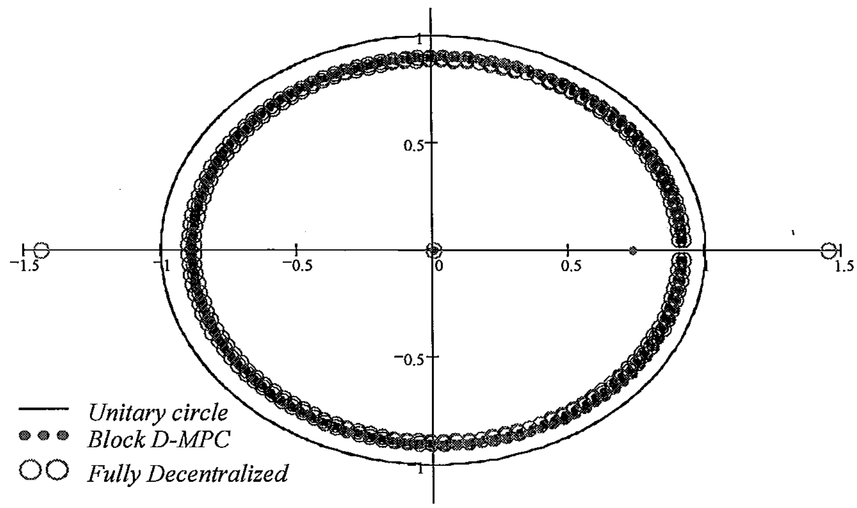


Figure 4.2 Unit Circle Analysis for Distillation Problem.

Figure 4.3 shows the closed-loop performances of centralized MPC (C-MPC), D-MPC and a fully decentralized MPC for a temperature change of $-1°C$ and $1°C$ on trays *21* and *7*, respectively.

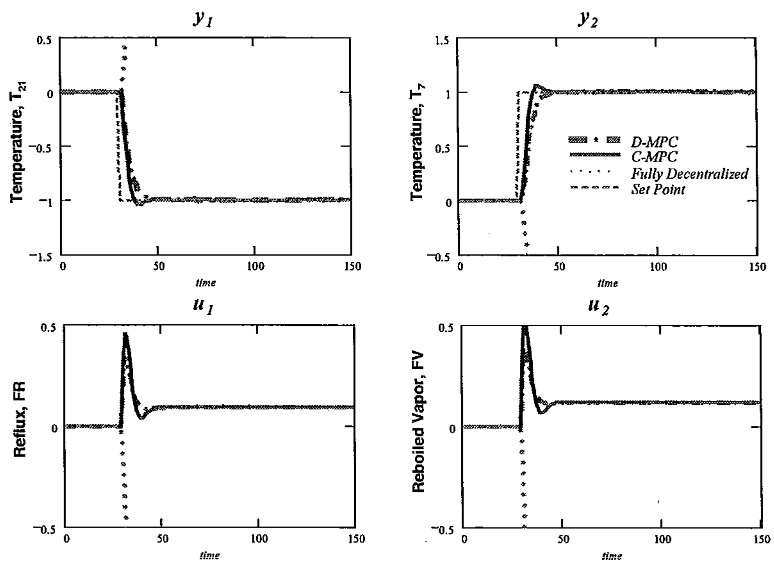


Figure 4.3 Closed Loop Performance for Distillation Problem.

It can be observed that the D-MPC not only is stable for a negative RGA configuration but it is also able to achieve acceptable performance. Additionally the D-MPC controller has a superior performance when compared to the independent block MPC (with no communication), which is unstable.

The D-MPC is also compared to another approach found in the literature, the Distributed MPC controller developed by Li et al. (2005). Basically, they propose an iterative approach to obtain a Nash equilibrium solution where the next iterate of the control action, Δu can be calculated as follows. Their approach allows distributed computation of the controller calculations, which is the chief difference from the current work.

$$\Delta u^{t+1} = D_0 \cdot \Delta u^t$$

For the two by two case D_0 is calculated as follows.

$$D_0 = \begin{pmatrix} 0 & -\left(A_{11}{}^T Q_1 A_{11} + R_1\right)^{-1} A_{11}{}^T Q_1 A_{12} \\ -\left(A_{22}{}^T Q_2 A_{22} + R_2\right)^{-1} A_{22}{}^T Q_2 A_{21} & 0 \end{pmatrix}$$

Thus in order to apply the iterative Distributed MPC the spectral radius must be less than one. This requirement will guarantee a convergent computation.

$$|\,\rho(D_0)| < 1$$

Using the parameters for the distillation example above the convergence condition results in a spectrum radius of $|\,\rho(D_0)| = 1.359$. This result indicates that the iterative method (Distributed MPC) presented by Li et al (2005) is not able to converge the algorithm for a negative RGA, much less to guarantee its closed loop stability. In addition, they have a stability criteria based on a contraction principle that guarantee nominal stability if and only if the norm of the eigenvalues is less than one.

$$\left\| eig\left(F - A_S \cdot \left(L \cdot K_{DMPC}\right) \cdot F\right) \right\| < 1$$

This stability criterion is more restrictive that the one presented in the Section 4.2.2.

Case 2 - Multivariable Blocks: The previous distillation column example showed the stability analysis of two *1x1* block controllers using the D-MPC approach. In order to show the stability criteria in a multi block system the next problem published by Lu (2001) is also analyzed. The nominal model is the following.

Block 1

$$\begin{pmatrix} y_{1_1} \\ y_{1_2} \end{pmatrix} = \begin{pmatrix} \frac{2}{16s^2 + 8s + 1} & \frac{1}{3s + 1} \\ \frac{1}{5s + 1} & \frac{3}{10s + 1} \end{pmatrix} \cdot \begin{pmatrix} u_{1_1} \\ u_{1_2} \end{pmatrix} + \begin{pmatrix} \frac{0.2}{10s + 1} & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} u_{2_1} \\ u_{2_2} \end{pmatrix}$$

Block 2

$$\begin{pmatrix} y_{2_1} \\ y_{2_2} \end{pmatrix} = \begin{pmatrix} \frac{2 \cdot e^{-61. \cdot s}}{75 \cdot s^2 + 20 \cdot s + 1} & \frac{-e^{-121 \cdot s}}{180 s^2 + 27 \cdot s + 1} \\ \frac{-0.4 e^{-57. \cdot s}}{100 s^2 + 25 \cdot s + 1} & \frac{2 \cdot e^{-117 \cdot s}}{240 s^2 + 32 \cdot s + 1} \end{pmatrix} \cdot \begin{pmatrix} u_{1_1} \\ u_{1_2} \end{pmatrix} + \begin{pmatrix} \frac{4e^{-5s}}{14s + 1} & \frac{2e^{-3s}}{22s + 1} \\ \frac{-1e^{-2s}}{25s + 1} & \frac{5}{27s + 1} \end{pmatrix} \cdot \begin{pmatrix} u_{2_1} \\ u_{2_2} \end{pmatrix}$$

This problem consists of two blocks each of which has a *two by two* MPC controller. The control configuration has a positive BRG. The base parameters used in the MPC controllers are shown in Table 4.2. Then, Figure 4.4 shows the closed loop simulation for set point changes using this set of tuning parameters. The dynamic response appears stable, which agrees with the stability analysis.

Table 4.2 Case 2 - Tuning Parameters

Block	*p*	*m*	*Q*	*R*
1	100	20	$\begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix}$
2			$\begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 30 & 0 \\ 0 & 30 \end{bmatrix}$

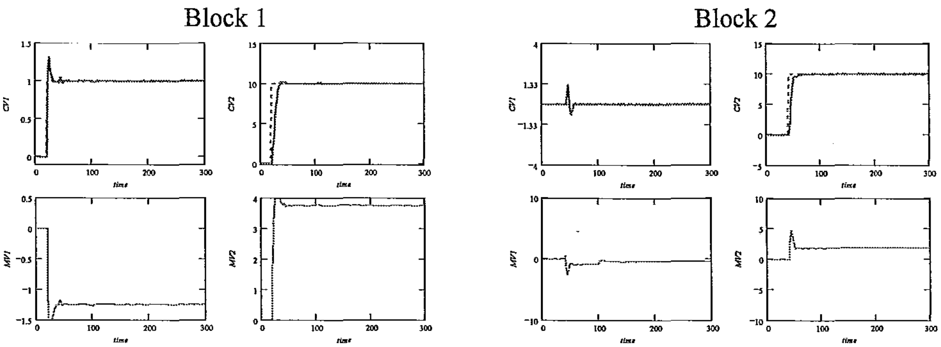


Figure 4.4 Closed Loop Simulation - Stable Solution

Next is a different case with a different set of tuning parameters where by trial and error and using the the stability expression in (4.21) it is possible to approximately detect a set of tuning parameters that results in having poles in the border of the unit circle as shown in Figure 4.5. For this case, the suppression factor for *Block 2* was modified to $R_2 = \begin{bmatrix} 5.25 & 0 \\ 0 & 5.25 \end{bmatrix}$.

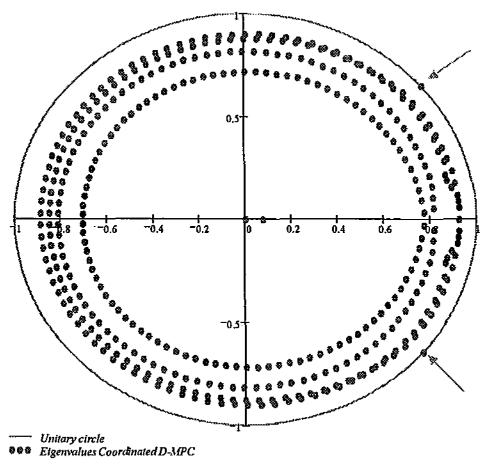


Figure 4.5 Unit Circle Analysis - Unstable Solution

Figure 4.6 illustrates the closed loop simulation subject to a couple of set point changes. The results confirm that the stability criterion is readily applicable for the control of multivariable blocks under the proposed D-MPC approach.

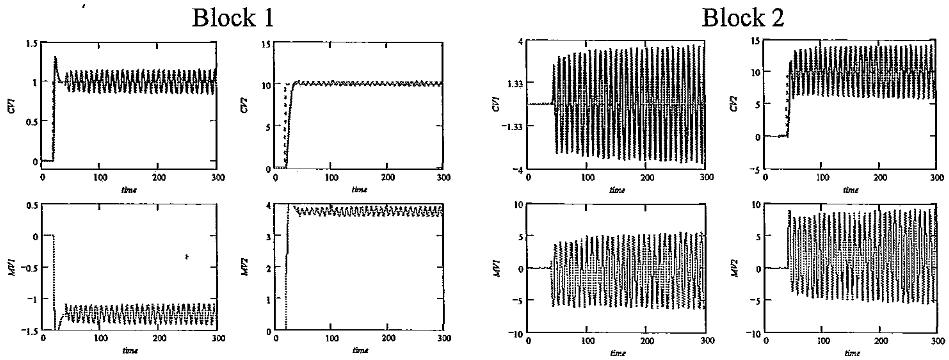


Figure 4.6 Closed Loop Simulation – Unstable Solution

The main result of this section is a stability analysis that is more general and less restrictive than the method published by Li et. al. (2005). The analysis of the closed loop stability was demonstrated for two different processes in which various control structures were defined through the controller matrix $(L \cdot K_{DMPC})$.

In addition, the D-MPC controller developed in this research was found to have a wider range of applicability than the Distributed D-MPC (Li et al, 2005) and the communication MPC used by Venkat and Rawlings (2004) for cases with negative RGA and BRG configurations. Furthermore, the results shows that the D-MPC controller presents a major advantage over the conventional block MPC strategy currently used in industrial practice which is not able to stabilized the plant paired on negative RGA or BRG configurations.

4.3 Existence and Stability Analysis of Single Horizon D-MPC

The previous sections in this chapter provide general results for the controllability and stability of D-MPC. In this section, we present additional analysis for a simple process structure to gain insight into some perhaps unexpected results that demonstrate that care must be taken in the design of D-MPC controllers. The analysis considers very simple systems that facilitate algebraic relationships for solution existence and stability.

Here, we analyze the D-MPC for a specific case known as Single Horizon MPC (Li, 2005), which reaches steady-state in one MPC controller execution as depicted in Figure 4.7. More specifically a *two by two* case and a *three by three* case under a multiple SISO control structure are analyzed in detail. Both processes are steady-state controllable. The objective of this analysis is to identify the control design requirements that guarantee the existence and uniqueness of the D-MPC solution.

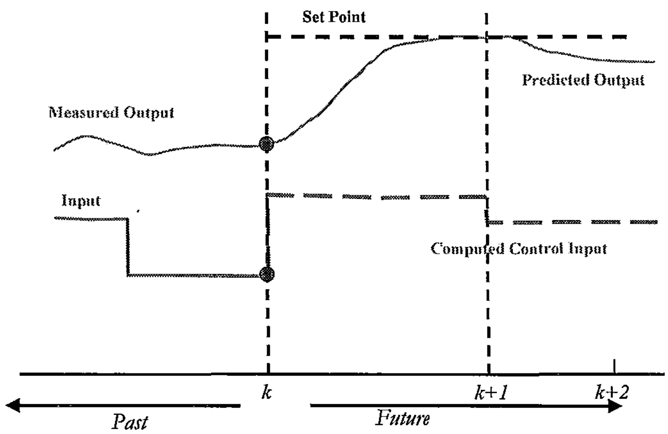


Figure 4.7 Single Horizon D-MPC

As presented in Chapter 3 the unconstrained D-MPC controller gain, (4.26) is a system of linear equations; therefore, the mathematical requirement in order to have a solution is to have a non-singular matrix, A_{DMPC}.

$$\begin{aligned} K_{DMPC} &= A_{DMPC}^{-1} \cdot b_{DMPC} \\ A_{DMPC} &= \left(A_D^T Q\, A_C + R\right) \\ b_{DMPC} &= A_D^T Q \end{aligned} \tag{4.26}$$

4.3.1 Existence of Single Horizon D-MPC Applied to Controllable Process

The analysis presented in this section covers first a *two by two* case and then a *three by three* case. Basically, matrix A_{DMPC} in (4.26) is expanded to investigate the existence and uniqueness of the D-MPC controller solution, and then by means of algebraic manipulations an expression that maps the singularity of the A_{DMPC} matrix for different tuning parameters is obtained. The results will demonstrate that for these two specific cases, the D-MPC controller always obtains a unique solution as long as the process is controllable and paired on a positive relative gain. However, for cases with negative (block) relative gains, the controller can be singular, and for non-singular controllers the feedback system can be nominally unstable; these are important new results.

For the sake of simplicity we start the analysis with the same *two by two* system shown in Section 4.2.3. In order to guarantee a unique solution for D-MPC the determinant of matrix A_{DMPC} must be nonzero. The single horizon controller has an input and output horizon of one $(m, p = 1)$ and the dynamic model consists only of steady state gains. The matrix A_{DMPC} is calculated as follows.

$$A_{DMPC} = \begin{pmatrix} Kp_{11}Q_1Kp_{11} + R_1 & Kp_{11}Q_1Kp_{12} \\ Kp_{21}Q_2Kp_{22} & Kp_{22}Q_2Kp_{22} + R_2 \end{pmatrix} \tag{4.27}$$

The analysis of the D-MPC controller for controllable systems (see Section 4.1.1) requires the next two expressions to be nonzero.

$$|A_{DMPC}| = (Kp_{11}Q_1Kp_{11} + R_1)(Kp_{22}Q_2Kp_{22} + R_2) - (Kp_{21}Q_2Kp_{22})(Kp_{11}Q_1Kp_{12}) \tag{4.28}$$

$$|Kp| = Kp_{11} \cdot Kp_{22} - Kp_{21} \cdot Kp_{12} \tag{4.29}$$

The first expression is the determinant of the matrix A_{DMPC} in equation (4.26). Having a nonzero determinant guarantees the existence and uniqueness of the solution. It is important to mention that the tuning parameters Q_f and R_f $(f = 1,2)$ can only take positive and nonnegative values, respectively. The second expression is the determinant of the steady state gain matrix of the system, which is nonzero for controllable systems. Since we have restricted the application of D-MPC to only controllable processes, no analysis is presented for non-controllable processes.

Expanding the D-MPC determinant (4.28) and setting it equal to zero results in the following.

$$\begin{aligned}(Kp_{11}Q_1Kp_{11}Kp_{22}Q_2Kp_{22})+(Kp_{11}Q_1Kp_{11}R_2)+(Kp_{22}Q_2Kp_{22}R_1)+R_1R_2\\ -(Kp_{21}Q_2Kp_{22})(Kp_{11}Q_1Kp_{12})=0\end{aligned} \quad (4.30)$$

Several cases are now analyzed with the objective of determining under which conditions equation (4.30) holds true. From (4.30) and dividing by the next expression:

$$(Kp_{11}Q_1Kp_{22}Q_2) \quad (4.31)$$

The following equation is obtained:

$$(Kp_{11}Kp_{22}-Kp_{12}Kp_{21})+\left(\frac{Kp_{11}R_2}{Kp_{22}Q_2}+\frac{Kp_{22}R_1}{Kp_{11}Q_1}+\frac{R_1R_2}{Kp_{11}Q_1Kp_{22}Q_2}\right)=0 \quad (4.32)$$

The objective of this analysis is to find if there exists a combination of Q_f and R_f $(f=1,2)$ that will make equation (4.32) hold true. If no combination exists then the D-MPC guarantees a unique solution for the entire set of tuning parameter values. This equation can be conveniently expressed in the following form.

$$C+D=0 \quad (4.33)$$

where

$$C=(Kp_{11}Kp_{22}-Kp_{12}Kp_{21}) \qquad D=\left(\frac{Kp_{11}R_2}{Kp_{22}Q_2}+\frac{Kp_{22}R_1}{Kp_{11}Q_1}+\frac{R_1R_2}{Kp_{11}Q_1Kp_{22}Q_2}\right) \quad (4.34)$$

From (4.33) the following can be observed:

- Terms in C can have different signs.
- All terms in D have the same sign.

Several cases are now analyzed based on equation (4.33)

Case 1: Zero Suppression Factors $Q_f > 0$ and $R_f = 0$.

The common case of tuning in practice considers positive values of the parameters; however, zero move suppression tuning, $R_f = 0$ can also be found in practice. In this case the term D in the determinant equation (4.33) becomes zero.

For the controller to be singular, the following expression must hold.

$$(Kp_{11}Kp_{22} - Kp_{12}Kp_{21}) = 0 \tag{4.35}$$

Equation (4.35) conflicts with the definition of a steady-state controllable process. Therefore, this equation will never hold for processes we have defined acceptable, and the D-MPC controller guarantees a unique solution for a controllable process with $R_f = 0$ and any positive combination of Q_f. This result applies to systems paired on either positive or negative RGA.

Case 2: Weighting factors $Q_f = 0$ and $R_f > 0$.

From (4.30) and by setting $Q_f = 0$ the following expression is obtained.

$$R_1 \cdot R_2 = 0 \tag{4.36}$$

This situation is not possible because in this case only positive values of R_f are considered. A trivial unique solution always exists for this case. We also note that setting Q_f to zero turns off the controller; therefore, this case is of little practical importance.

Case 3: One-way interaction.

In this case one of the interaction models is zero, $(Kp_{12}\,Kp_{21} = 0)$.

$$(Kp_{11}Kp_{22}) + \left(\frac{Kp_{11}R_2}{Kp_{22}Q_2} + \frac{Kp_{22}R_1}{Kp_{11}Q_1} + \frac{R_1R_2}{Kp_{11}Q_1Kp_{22}Q_2}\right) = 0 \tag{4.37}$$

Since the sign of $(Kp_{11} \cdot Kp_{22})$ is the same as the sign of $\left({Kp_{11}}/{Kp_{22}}\right)$ both terms C and D have the same sign. This guarantees that the determinant is never zero, and therefore, the equation never holds and a unique solution always exists. This result also applies to systems paired on either positive or negative RGA.

Case 4: Two-way interaction Paired on Positive RGA

The RGA for a two by two system paired on $(u_1 - y_1)$ and $(u_2 - y_2)$ can be calculated as following.

$$\Lambda_{11} = \Lambda_{22} = \left(\frac{Kp_{11}Kp_{22}}{Kp_{11}Kp_{22} - Kp_{12}Kp_{21}}\right) \tag{4.38}$$

If the RGA is positive this means that the sign of the denominator and numerator must be the same. The positive RGA guarantees that both terms C and D in equation (4.33) have the same sign. This condition guarantees the existence of a unique solution for the D-MPC controller.

Case 5: Two-way interaction Paired on Negative RGA

The RGA expression for the same system paired on $(u_1 - y_1)$ and $(u_2 - y_2)$ and having a negative RGA is the following.

$$\Lambda_{11} = \Lambda_{22} = \left(\frac{Kp_{11}Kp_{22}}{Kp_{11}Kp_{22} - Kp_{12}Kp_{21}} \right) < 0 \qquad (4.39)$$

In order to have a negative RGA the denominator and numerator in equation (4.39) must have opposite signs. This condition indicates terms C and D in equation (4.33) also have opposite signs. In this situation there are multiple combinations of the tuning parameters that would make the D-MPC controller matrix singular.

The next step is to identify and set the limits of the region where a singularity in the D-MPC gain matrix is possible. An expression of the ratio of tuning parameters that makes A_{DMPC} singular is obtained from the determinant expression (4.28). The terms can be rearranged in the following way:

$$\left(\frac{Kp_{11}}{Kp_{12}} + \frac{\left[R_1 / Q_1 \right]}{Kp_{11}Kp_{12}} \right)^{-1} = \frac{Kp_{22}}{Kp_{21}} + \frac{\left[R_2 / Q_2 \right]}{Kp_{21}Kp_{22}} \qquad (4.40)$$

An expression that maps the tuning combinations that make A_{DMPC} matrix singular is then obtained:

$$\left[R_2 / Q_2 \right] = \frac{Kp_{12}Kp_{11}Kp_{21}Kp_{22}}{Kp_{11}^2 + \left[R_1 / Q_1 \right]} - Kp_{22}^2 \qquad (4.41)$$

Some observations can be made from expression (4.41)

- The extreme points of this singularity *line* (See Figure 4.8) are obtained by evaluating expression (4.41) for $\left[R_1 / Q_1 \right] = 0$ and $\left[R_2 / Q_2 \right] = 0$ respectively. Through algebraic manipulations these extreme points, which are now called upper bounds, are found to be related to the RGA as follows.

$$\left[{R_f}/{Q_f}\right]_{UB} = -\frac{\left(Kp_{ff}\right)^2}{A_{ff}} \tag{4.42}$$

- Once outside the bounds of the singularity *line* the required value of $\left[{R_f}/{Q_f}\right]$ that will produce a singular A_{DMPC} matrix would have to be negative which is excluded by tuning guidelines.
- Inside the bounds just defined only the combination of tuning parameters that satisfies expression (4.41) produce a singular A_{DMPC} matrix.

The results are summarized in Table 4.3. It is important to again note that in order to have a workable system the values of the tuning parameter Q must be positive. Otherwise, if Q_1 and $Q_2 = 0$ the system is undefined, and a trivial solution is achieved.

Table 4.3 Uniqueness of Single Horizon D-MPC for Controllable Processes

Tuning \ RGA	Two-way interaction: Negative RGA	Two-way interaction: Positive RGA	One-way Interaction
$R_f = 0,\ Q_f > 0$	$\lvert A_{DMPC}\rvert \neq 0$, Unique Solution	$\lvert A_{DMPC}\rvert \neq 0$, Unique Solution	$\lvert A_{DMPC}\rvert \neq 0$, Unique Solution
$R_f > 0,\ Q_f = 0$			
$R_f > 0,\ Q_f > 0$	For some tuning combinations $\lvert A_{DMPC}\rvert = 0$, ***Singular Controller**		

$f = 1,2$

If the *two by two* process is steady-state controllable, and it is paired on positive RGA then the D-MPC controller is able to guarantee a unique solution for any combination of the tuning parameters. On the other hand for processes with two-way interaction, and loops paired on a negative RGA a singularity will appear under certain circumstances.

The results are illustrated in Figure 4.8 for a negative RGA configuration. These results also confirm that for positive RGA configuration no singularity region exists, since the upper bounds would be negative and the ratio $\left[{R_f}/{Q_f}\right]$ can only take nonnegative values.

This figure could be calculated for any process by evaluating expression (4.41) for different values of $\left[{R_1}/{Q_1}\right]$ and then plotting the corresponding $\left[{R_2}/{Q_2}\right]$.

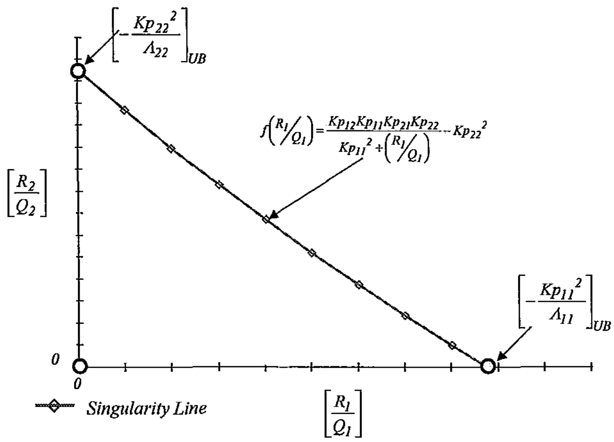


Figure 4.8 Singularity Line - Single Horizon D-MPC (Negative RGA)

It is important to note that the results presented here have as main conditions that **(1)** the process is controllable and **(2)** each subsystem must be controllable.

A similar treatment was performed on a *three by three* system under a multiple SISO control structure, a brief presentation is shown next and the results are then summarized. The A_{DMPC} matrix for the three by three system is the following. Here it is important to mention that the block relative gain (BRG) for multiple blocks of SISO controllers is the same as the RGA.

$$A_{DMPC} = \begin{pmatrix} Kp_{11}Q_1Kp_{11} + R_1 & Kp_{11}Q_1Kp_{12} & Kp_{11}Q_3Kp_{13} \\ Kp_{22}Q_2Kp_{21} & Kp_{22}Q_2Kp_{22} + R_2 & Kp_{22}Q_3Kp_{23} \\ Kp_{33}Q_3Kp_{31} & Kp_{33}Q_3Kp_{32} & Kp_{33}Q_3Kp_{33} + R_3 \end{pmatrix} \qquad (4.43)$$

The procedure again is to expand the determinant and find and expression that relate the RGA of the system to the bounds of singularity region. The algebraic analysis begins with the following partition of the matrix.

$$\dot{A} = \begin{pmatrix} Kp_{11}Q_1Kp_{11} + R_1 & Kp_{11}Q_1Kp_{12} \\ Kp_{21}Q_2Kp_{22} & Kp_{22}Q_2Kp_{22} + R_2 \end{pmatrix} \qquad \dot{B} = \begin{pmatrix} Kp_{11}Q_1Kp_{13} \\ Kp_{22}Q_2Kp_{23} \end{pmatrix}$$
$$\dot{C} = \begin{pmatrix} Kp_{33}Q_3Kp_{31} & Kp_{33}Q_3Kp_{32} \end{pmatrix} \qquad \dot{D} = \left(Kp_{33}Q_3Kp_{33} + R_3\right) \tag{4.44}$$

The Schur complement is now used to calculate the determinant.

$$|A_{DMPC}| = |\dot{D}|\left|\dot{A} - \dot{B} \cdot \dot{D}^{-1} \cdot \dot{C}\right| \tag{4.45}$$

Considering that the first term $|\dot{D}|$ is always invertible for a controllable process, the proof for a nonzero determinant of A_{DMPC} focuses on the determinant of the Schur complement, $|Sc| = \left|\dot{A} - \dot{B} \cdot \dot{D}^{-1} \cdot \dot{C}\right|$. A general expression that maps the singularity of the D-MPC controller is now obtained by equating the Schur complement to zero.

$$\left[\frac{R_f}{Q_f}\right] = \frac{-Kp_{ff} \cdot \left[|Kp| + \frac{|\dot{M}_{jj}|}{Kp_{jj}} \cdot \frac{R_j}{Q_j} + \frac{|\dot{M}_{kk}|}{Kp_{kk}} \cdot \frac{R_k}{Q_k} + \frac{Kp_{jj}}{Kp_{jj} \cdot Kp_{kk}} \cdot \frac{R_j}{Q_j} \cdot \frac{R_k}{Q_k}\right]}{\left[|\dot{M}_{ff}| + \frac{Kp_{jj}}{Kp_{kk}} \cdot \frac{R_k}{Q_k} + \frac{Kp_{kk}}{Kp_{jj}} \cdot \frac{R_j}{Q_j} + \frac{1}{Kp_{jj} \cdot Kp_{kk}} \cdot \frac{R_j}{Q_j} \cdot \frac{R_k}{Q_k}\right]} \tag{4.46}$$

Where Kp is the steady state gain matrix and $\dot{M}_{jj}$, $\dot{M}_{kk}$ and $\dot{M}_{ff}$ are the minors of elements Kp_{jj}, Kp_{kk} and Kp_{ff} respectively (i.e. $\dot{M}_{jj}$ is formed by deleting row j and column j of Kp).

The extreme points also called upper bounds of this singularity *expression* are obtained by evaluating expression (4.46) for $\left[{R_j}/{Q_j}\right] = 0$ and $\left[{R_k}/{Q_k}\right] = 0$.

$$\left[\frac{R_f}{Q_f}\right]_{UB} = -Kp_{ff}\frac{|Kp|}{|\dot{M}_{ff}|} \tag{4.47}$$

Finally the RGA for a *three by three* system is calculated as follows.

$$\Lambda_{ff} = Kp_{ff} \frac{|\dot{M}_{ff}|}{|Kp|} \tag{4.48}$$

Therefore, it is found that the bounds on the singularity line are again related to the RGA in the following way.

$$\left[\frac{R_f}{Q_f}\right]_{UB} = -\Lambda_{ff}\left(\frac{|Kp|}{|\dot{M}_{ff}|}\right)^2 \tag{4.49}$$

Again if a positive RGA $(\Lambda_{ff} > 0)$ is considered then the bound will fall in the negative part of the tuning spectrum. On the other hand a negative RGA $(\Lambda_{ff} < 0)$ will have at least three tuning combinations that will make the controller singular.

4.3.2 Stability of the Single Horizon D-MPC

Once the existence of the controller is analysed the next step is to analyze the nominal stability (no model mismatch) of D-MPC. This section applies the previous results for nominal stability of multivariable controllers; the analysis is tailored to (a) multiple blocks SISO control and (b) Single Horizon D-MPC controller.

The closed loop system depicted in Figure 4.9 is considered.

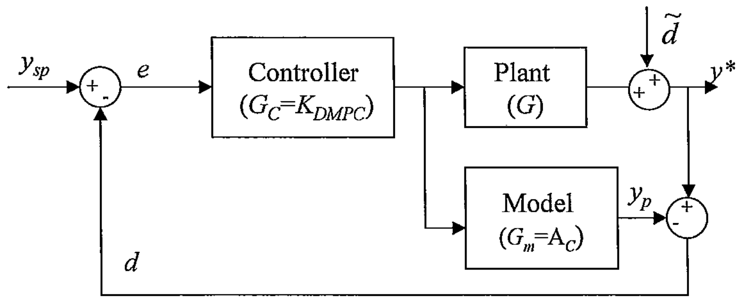


Figure 4.9 Closed Loop System

The D-MPC controller is defined as follows; the formulation details are presented in Section 3.2.

$$K_{DMPC} = \left(A_D{}^T Q\ A_C + R\right)^{-1} A_D{}^T Q \tag{4.50}$$

This controller is built from dynamic matrices (steady-state gains for this case) and tuning parameters. The state-space representation of the step weight model for the single input single output dynamic system is the following.

$$X(k) = F \cdot X(k-1) + A_S \cdot \Delta u_{(k-1)}$$
$$X(k) = [y_0(k) \quad y_1(k) \quad \cdots \quad y_{p-1}(k)] \tag{4.51}$$

Again $X(k)$ denotes the p predicted states starting from the value at time k, and $\Delta u_{(k-1)}$ is the manipulated variable change at time $k-1$. The step response coefficients are contained in A_S. For the case of single step horizon $(p,m=1)$ where the plant reaches steady-state in one execution the prediction model is reduced to:

$$X(k+1) = 1 \cdot X_1(k) + a_p \cdot \Delta u \tag{4.52}$$

Where a_p corresponds to a steady-state gain Kp.

In this prediction the steady-state effect of the input change is added to the current predicted value. For a multivariable system with M inputs and P outputs the resulting prediction model will have the following form.

$$\begin{bmatrix} X(k+1)_1 \\ X(k+1)_2 \\ \vdots \\ X(k+1)_P \end{bmatrix} = \begin{bmatrix} [1] & 0 & 0 & 0 \\ 0 & [1] & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & [1] \end{bmatrix} \cdot \begin{bmatrix} X(k)_1 \\ X(k)_2 \\ \vdots \\ X(k)_P \end{bmatrix} + \begin{bmatrix} Kp_{11} & Kp_{12} & \cdots & Kp_{1M} \\ Kp_{21} & Kp_{22} & \cdots & Kp_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ Kp_{P1} & Kp_{P2} & \cdots & Kp_{PM} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \Delta u_M \end{bmatrix} \tag{4.53}$$

$$X(k+1) = F \cdot X(k) + A_S \cdot \Delta u$$

The elements Kp_{gf} are the steady-state gains, which correspond to element $a_{P_{gf}}$ of the corresponding step response model. The state space formulation for the D-MPC controller is defined by the following closed loop dynamic equation.

$$\begin{aligned} X(k) &= \left[F - S \cdot \left(L \cdot K_{DMPC}\right) \cdot F\right] \cdot X(k-1) + \left[A_S \cdot L \cdot K_{DMPC}\right] \cdot X_{sp}(k-1) \\ y(k) &= C \cdot X(k) \end{aligned} \tag{4.54}$$

Where K_{DMPC} is the control gain matrix defined in (4.50).

In this way the poles of the following expression define the stability of the closed loop system.

$$H_{SS} = \left[F - A_S \cdot \left(L \cdot K_{DMPC}\right) \cdot F\right] \tag{4.55}$$

For this single horizon case the matrices, F, A_S and L are defined as follows.

$$F = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \text{ and } F \in \Re^{P \times P} \tag{4.56}$$

And A_S contains the only the steady-state gains $Kp_{g,f}$ arranged in the following way.

$$A_S = \begin{pmatrix} Kp_{1,1} & \cdots & Kp_{1,M} \\ \vdots & \ddots & \vdots \\ Kp_{P,1} & \cdots & Kp_{P,M} \end{pmatrix} \tag{4.57}$$

Finally $\hat{L}$ also becomes an identity matrix.

$$\hat{L} = \begin{pmatrix} L_0 & O & O \\ O & \ddots & O \\ O & O & L_0 \end{pmatrix} \quad \begin{matrix} L_0 = [1] \\ O = [0] \end{matrix}. \tag{4.58}$$

Since both F and $\hat{L}$ are identity matrices the expression that analyzes the closed loop stability of a Single Horizon D-MPC can be simplified as follows.

$$H_{SS} = [I - A_S \cdot K_{DMPC}] \tag{4.59}$$

Where $I \in \Re^P$ is an identity matrix.

A stable plant under a D-MPC that reaches steady-state in one execution period is stable if the eigenvalues of the matrix H_{SS} defined in expression (4.59) lie inside the unit circle.

4.3.3 Relationship between RGA and Nominal Stability Regions

In this section a specific case is considered, a *two by two* system is analyzed to explicitly delimit their stability regions under a Single Horizon D-MPC controller. Then, some conclusions are drawn relating negative RGA configurations to unstable regions.

The analysis of the *two by two* case begins by evaluating the following equation.

$$H_{SS} = [I - A_C \cdot K_{DMPC}] \tag{4.60}$$

All the matrices involved have a *two by two* dimension $\left(\Re^{2\times 2}\right)$. It is also observed that A_S has been replaced with A_C, which for single horizon MPC controllers consists merely of steady-state gains.

The next step in the stability analysis is to substitute the D-MPC controller, K_{DMPC}.

$$H_{SS} = I - A_C\left[\left(A_D{}^T Q\ A_C + R\right)^{-1} A_D{}^T Q\right] \tag{4.61}$$

For this *two by two* case the stability (boundary) line that divides the stable and unstable regions (See Figure 4.10) can be found by setting one of the eigenvalues to negative one (i.e. $v_1 = -1$) and then by using the following eigenvalues properties obtain the tuning parameter that will produce continuous oscillatory response.

$$\sum_{f=1}^{M} v_f = tr(H_{SS}) \qquad \prod_{f=1}^{M} v_f = |H_{SS}| \tag{4.62}$$

Where M is the number of input variables.

In this way for a fixed value of $\left[{}^{R_1}\!/_{Q_1}\right]$ the corresponding value of $\left[{}^{R_2}\!/_{Q_2}\right]$ on the stability limit can be obtained by solving the next set of nonlinear equations.

$$(-1)+v_2 = tr\left[H_{SS}\left\{{}^{R_2}\!/_{Q_2}\right\}\right] \qquad v_2 = -\left|H_{SS}\left\{{}^{R_2}\!/_{Q_2}\right\}\right|$$

Where $H_{SS}\left\{{}^{R_2}\!/_{Q_2}\right\}$ indicates H_{SS} as function of $\left\{{}^{R_2}\!/_{Q_2}\right\}$. The solution of this set of equations provides the entire stability border.

Figure 4.10 shows the boundary line that delimits the region for stability for a D-MPC controller with negative RGA configuration. Additionally the singularity line described in Section 4.3.1-Case 5, which contains the tuning combinations that make the controllers singular is also illustrated.

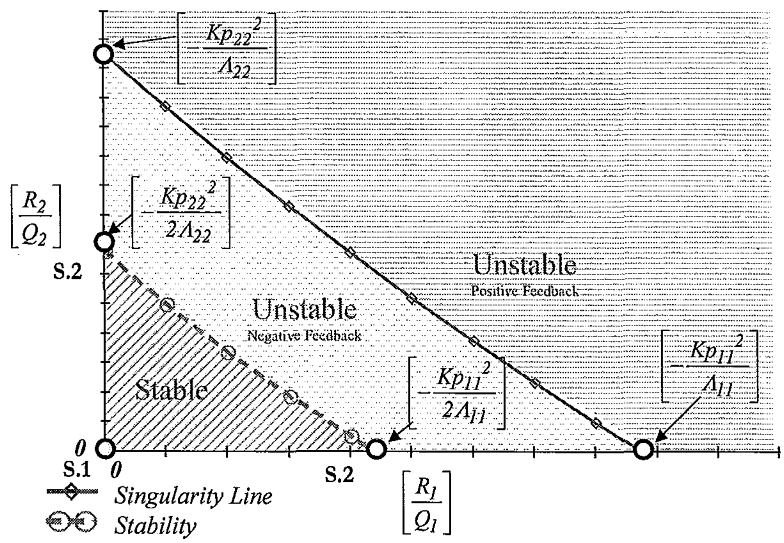


Figure 4.10 Stability Regions for a 2x2 Single Horizon MPC (Negative RGA)

The region defined between the stability line and the singularity line results in an unstable behaviour with negative feedback. The points within this region have a maximum eigenvalue that lies outside the unit circle. Then, on the singularity line the matrix H_{SS}

becomes singular and the K_{DMPC} controller has no solution. Beyond this singularity line the matrix H_{SS} becomes nonsingular but now the behaviour is unstable with a positive feedback.

Figure 4.10 also points out the main tuning combinations that delimit the stability area (**S.1** and **S.2**) which are described as follows.

S.1 It can be observed that the stable region will include the case where the suppression factor parameter is at its lower bound ($R = 0$). This in fact is the lower bound of the region; in this case matrix H_{SS} becomes a zero matrix with eigenvalues equal to zero.

$$H_{SS} = I - A_C\left(A_D{}^T Q\, A_C\right)^{-1} A_D{}^T Q$$
$$H_{SS} = I - I = 0 \tag{4.63}$$

S.2 The tuning combination that divides the unstable and stable regions on the axis can be easily obtained. First set one of the tuning parameter to zero (i.e. $\left[{R_1}/{Q_1}\right] = 0$). The resulting H_{SS} matrix is the following.

$$H_{SS} = \begin{bmatrix} 0 & 0 \\ -\left(\dfrac{Kp_{21}}{Kp_{11}}\right).\Psi & \Psi \end{bmatrix}$$

Where

$$\Psi = \left(\frac{1}{Kp_{22}^2}\left(\frac{R_2}{Q_2}\right)\cdot\left(\frac{1}{\Lambda_{22}} - \frac{1}{Kp_{22}^2}\left(\frac{R_2}{Q_2}\right)\right)^{-1}\right)$$

Where Λ_{22} is the relative gain. This in turn produces a zero determinant $|H_{SS}| = 0$. The next step is to set one eigenvalue to negative one (i.e. $v_1 = -1$) and then using the eigenvalue properties in (4.62) the following is obtained.

$$(-1)\cdot v_2 = |H_{SS}| = 0$$

$$(-1)+v_2 = tr\left[H_{SS}\left\{{R_2}/{Q_2}\right\}\right] = \Psi$$

From the first equation $v_2 = 0$, then the second equation is.

$$-1+0 = \left(\frac{1}{Kp_{22}^2}\left(\frac{R_2}{Q_2}\right)\cdot\left(\frac{1}{\Lambda_{22}} - \frac{1}{Kp_{22}^2}\left(\frac{R_2}{Q_2}\right)\right)^{-1}\right)$$

Then by performing algebraic manipulations a general expression for the tuning combination that divides the unstable and stable regions on the axis is obtained.

$$\left[\frac{R_f}{Q_f}\right]\Bigg|_{UB} = -\left(\frac{1}{2}\right)\frac{Kp^2_{ff}}{\Lambda_{ff}} \qquad f = 1,2 \tag{4.64}$$

From expression (4.64) it can be concluded that a *two by two* process paired on positive RGA cannot become unstable due to tuning. This is because the limits of the stability region are related to the relative gain $(\Lambda_{f,f})$.

$$\begin{array}{ll} if\ \ \Lambda_{f,f} < 0 & \left[\frac{R_f}{Q_f}\right]\Bigg|_{UB} = -\left(\frac{1}{2}\right)\frac{Kp^2_{ff}}{\Lambda_{ff}} > 0 \\ if\ \ \Lambda_{f,f} > 0 & \left[\frac{R_f}{Q_f}\right]\Bigg|_{UB} = -\left(\frac{1}{2}\right)\frac{Kp^2_{ff}}{\Lambda_{ff}} < 0 \end{array} \tag{4.65}$$

The previous statement proves that this specific Single Horizon D-MPC controller is stable for systems with integrity.

4.3.4 The Shell Standard Control Problem (SSCP), a three by three case:

In order to test the previous result on a larger case a single horizon controller was used for the Shell Standard control problem, which is later described in Section 5.4. The nominal model considers only the steady state gains.

$$\begin{pmatrix} x_{TD} \\ x_{SD} \\ T_{BR} \end{pmatrix} = \begin{pmatrix} 4.05 & 1.77 & 5.88 \\ 5.39 & 5.72 & 6.9 \\ 4.38 & 4.42 & 7.20 \end{pmatrix} \cdot \begin{pmatrix} F_{TD} \\ F_{SD} \\ F_{BR} \end{pmatrix} + \begin{pmatrix} 1.2 & 1.44 \\ 1.52 & 1.83 \\ 1.14 & 1.26 \end{pmatrix} \cdot \begin{pmatrix} F_{IR} \\ F_{UR} \end{pmatrix} \tag{4.66}$$

The BRG that corresponds exactly to the RGA is the following.

$$\Lambda = \begin{pmatrix} 2.08 & \langle -0.73 \rangle & -0.35 \\ 3.42 & 0.93 & \langle -3.36 \rangle \\ \langle -4.5 \rangle & 0.79 & 4.71 \end{pmatrix} . \tag{4.67}$$

In order to test the stability results for negative RGA presented above, the control structure was intentionally chosen with the following pairings, $(x_{TD} - F_{SD})$, $(x_{SD} - F_{BR})$ and $(T_{BR} - F_{TD})$. For convenience and in order to use a $(y_f - u_f)$ pairing configuration the input and output variables are named as follows $(u_1 = F_{SD},\ u_2 = F_{BR},\ u_3 = F_{TD})$ and $(y_1 = x_{TD},\ y_2 = x_{SD},\ y_3 = T_{BR})$. The resulting model can be expressed as follows.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \langle 1.77 \rangle & 5.88 & 4.05 \\ 5.72 & \langle 6.9 \rangle & 5.39 \\ 4.42 & 7.2 & \langle 4.38 \rangle \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{pmatrix} 1.2 & 1.44 \\ 1.52 & 1.83 \\ 1.14 & 1.26 \end{pmatrix} \cdot \begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \tag{4.68}$$

The idea is to apply the general expression described in (4.49) to obtain the bounds of the singularity line.

In a similar way as for the *two by two* case the extreme points (on the axis) of the line that divides the stable and unstable regions can be obtained by fixing two of the three tuning parameters to zero (i.e. $R_1/Q_1 = R_2/Q_2 = 0$). For this case the resulting H_{SS} matrix is the following.

$$H_{SS} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \dfrac{|\dot{M}_{13}|}{|\dot{M}_{33}|}\cdot\Psi & \dfrac{|\dot{M}_{23}|}{|\dot{M}_{33}|}\cdot\Psi & \Psi \end{bmatrix}$$

Where

$$\Psi = \left(\frac{|\dot{M}_{33}|}{Kp_{33}}\left(\frac{R_3}{Q_3}\right) \cdot \left(|Kp| + \frac{|\dot{M}_{33}|}{Kp_{33}}\left(\frac{R_3}{Q_3}\right)\right)^{-1} \right)$$

Then by fixing one eigenvalue to negative one (i.e. $v_1 = -1$) and using the eigenvalue properties in (4.62) the following is obtained.

$$(-1)\cdot v_2 \cdot v_3 = |H_{SS}| = 0$$

$$(-1) + v_2 + v_3 = tr\left[H_{SS}\left\{{R_3}/{Q_3}\right\}\right] = \Psi$$

From the first equation $v_2 \cdot v_3 = 0$, if we define $v_2 = v_3 = 0$ then the second equation is.

$$-1+0+0 = \left(\frac{|\dot{M}_{33}|}{Kp_{33}}\left(\frac{R_3}{Q_3}\right) \cdot \left(|Kp| + \frac{|\dot{M}_{33}|}{Kp_{33}}\left(\frac{R_3}{Q_3}\right)\right)^{-1} \right)$$

Finally the general expression that obtains the tuning that divides the unstable and stable regions on the axis can be obtained by repeating the same procedure for the rest of the tuning parameters.

$$\left(\frac{R_f}{Q_f}\right) = -\frac{\Lambda_{ff}}{2}\cdot\left(\frac{|Kp|}{|\dot{M}_{ff}|}\cdot\right)^2 \qquad (4.69)$$

Where Λ_{ff} is the corresponding RGA as defined in (4.48).

Finally Table 4.4 shows the tuning parameters that define the bounds on the axis of the singularity and stability regions for the Shell Problem.

Table 4.4 SSCP - Tuning Parameters, Upper Bounds

Block	Singularity Bounds $\left[\frac{R_f}{Q_f}\right]_{Singular} = -A_{ff}\left(\frac{\lvert Kp\rvert}{\lvert \dot{M}_{ff}\rvert}\right)^2$	Stability Bounds $\left[\frac{R_f}{Q_f}\right]_{Stable} = -\left(\frac{A_{ff}}{2}\right)\cdot\left(\frac{\lvert Kp\rvert}{\lvert \dot{M}_{ff}\rvert}\right)^2$
1	$\left[\frac{R_1}{Q_1}\right]_{Singular} = (4.298,0,0)$	$\left[\frac{R_1}{Q_1}\right]_{Stable} = (2.149,0,0)$
2	$\left[\frac{R_2}{Q_2}\right]_{Singular} = (0,14.17,0)$	$\left[\frac{R_2}{Q_2}\right]_{Stable} = (0,7.088,0)$
3	$\left[\frac{R_3}{Q_3}\right]_{Singular} = (0,0,4.263)$	$\left[\frac{R_3}{Q_3}\right]_{Stable} = (0,0,2.132)$

In order to test the stability bounds Figure 4.11 illustrates the unitary circle using the stability bound for *Block 1*. This tuning is then simulated subject to several set point changes.

Figure 4.12 illustrate the performance of the closed loop simulation for the Single Horizon D-MPC.

$$\begin{pmatrix} R_1 \\ R_2 \\ R_3 \end{pmatrix} = \begin{pmatrix} 2.149 \\ 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} Q_1 \\ Q_2 \\ Q_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{4.70}$$

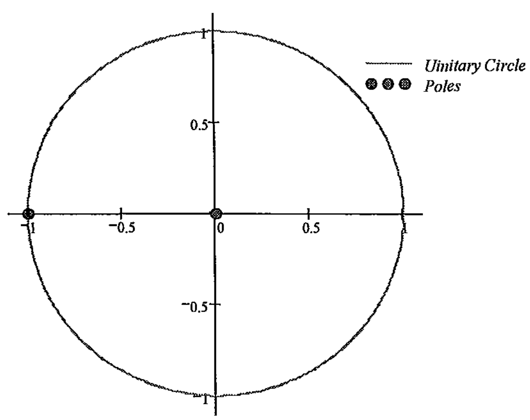


Figure 4.11 Unitary Circle, Single Horizon D-MPC

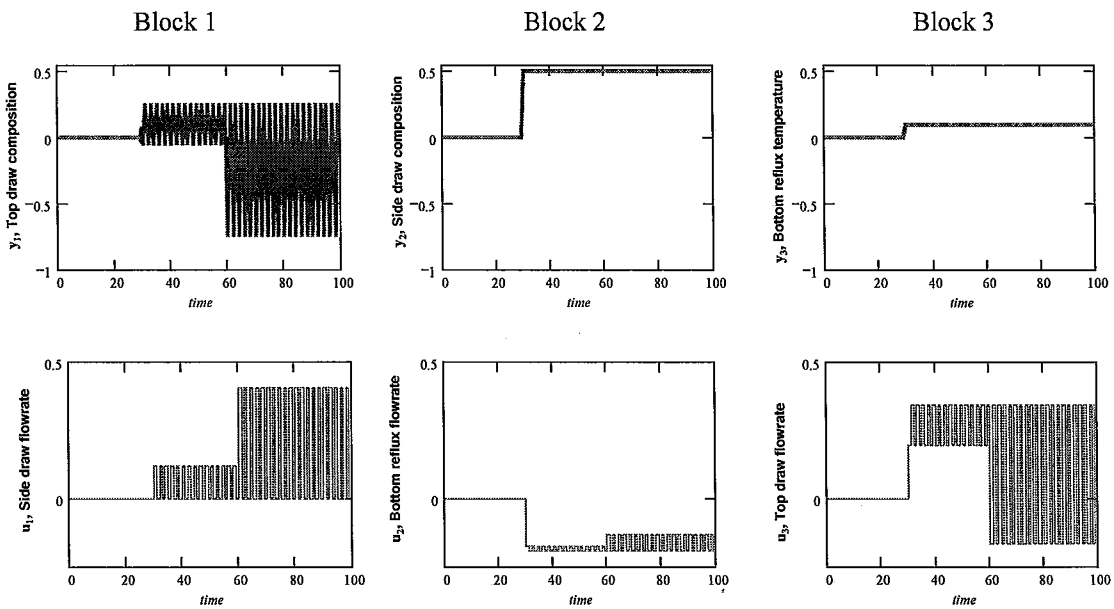


Figure 4.12 Closed Loop Simulation, Single Horizon D-MPC

Although no further analysis was done for larger systems, the results obtained provide some insight about the relationship between the RGA, the singularity line and the stability regions for Single Horizon D-MPC.

4.4 Application of Results for D-MPC Design and Tuning

Based on the analysis made in this chapter a basic methodology for the implementation of D-MPC controller can be developed. Figure 4.13 illustrates the sequential steps required in order to produce a non-singular and stable D-MPC controller.

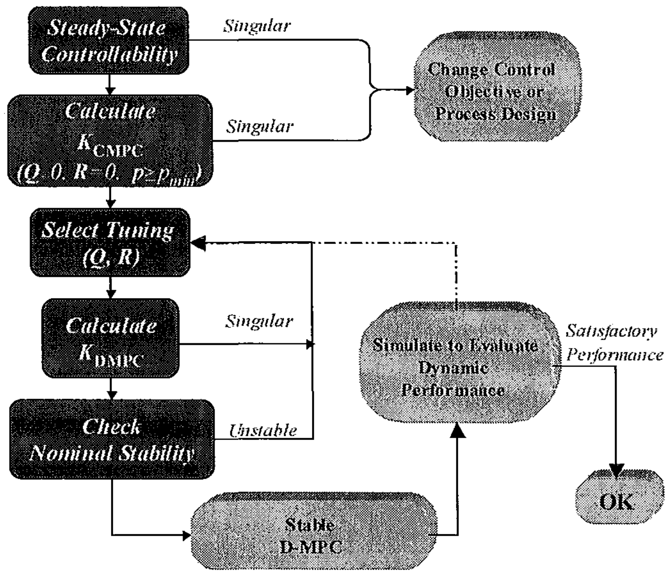


Figure 4.13 D-MPC Basic Steps of D-MPC Design Method

First, steady-state controllability and the existence of a centralized controller must be verified. If it is not possible to produce a nonsingular centralized controller then the D-MPC controller is not implemented. The next steps require adjusting the tuning parameters in order to produce a nonsingular and stable D-MPC controller. Finally, simulations can be made in order to fine tune the controller.

4.5 Summary and Conclusions

In this chapter the D-MPC formulation was analysed in order to define the requirements to guarantee the existence and uniqueness of a control solution. In the same way an important method to analyze closed loop stability was presented. Finally, several examples were analyzed including the Single Horizon D-MPC controller, for which interesting results relating matrix singularity and RGA were obtained. Some concluding remarks are now made based on the results obtained in this chapter.

D-MPC/C-MPC Controllability: It was decided that only stable processes that can be controlled by C-MPC control would be considered for D-MPC control. Since applications are for continuous processes that track their set points, the process must be steady-state controllable, as defined in this chapter, with the column rank of the gain matrix being equal to the number of controlled variables. Furthermore, by analyzing the structure of the C-MPC controller some guideline can be followed. It was found that prediction horizon of the controller influences the controllability, and a minimum output horizon was defined. Also, it was found that the dynamic matrix A_C alone defines the invertibility of the control matrix. No simple guideline or mathematical test less complex than evaluating the rank of A_C was found; however, the rank of A_C or of the controller gain matrix can be evaluated to ensure that a controller exists for the tuning selected.

D-MPC Stability: The D-MPC approach that is built from simultaneous optimality conditions results in a controller with a well-defined structure. Furthermore, this defined structure made it easy to apply a classic discrete time stability analysis (Lee et al 1994). A couple of cases with different block structures (SISO, MIMO) proved the successful applicability of this stability analysis. However it is important to note that only nominal processes and unconstrained controllers are considered.

Single Horizon D-MPC Controller: For this controller where the plant reaches steady state in one controller execution the following can be concluded: (1) D-MPC paired on positive RGA, have a unique stable solution for any controllable plant, $(|Kp| \neq 0)$. (2) On the other hand processes paired on negative RGA present both a stable (negative feedback) and an unstable (positive feedback) zone. This negative configuration may also present a singularity zone even for a controllable plant. A couple of numerical examples proved mathematically that at least on the extreme points (i.e. on the axis) the singularity line lies outside of the stability zone.

In summary the analysis presented in this chapter showed that certain D-MPC designs could be singular or nominally unstable depending on the control structure (BRG) and tuning. Since no general guarantees for existence and stability were derived, a design procedure was developed in order to guarantee a stable control system.

Chapter 5

Block D-MPC Performance

The fundamental approach to block MPC and control algorithms developed in Chapter 3 and the stability analysis developed in Chapter 4 are applied to several process control applications in this chapter. The cases have been selected to evaluate the ease and generality of application of D-MPC and to compute the dynamic performance achieved by D-MPC in comparison to centralized MPC and independent block MPC.

In this section, we demonstrate the D-MPC controller first on a *two by two* distillation column example with the purpose of showing the main advantages of the controller. Then a benchmark process, the Shell Standard Control Problem and a *four by four* fire heater system are used to demonstrate how the method handles different control structures and multiple interactions. Several design configurations with different integrity (positive, negative and zero BRG) are evaluated.

5.1 Defining Independent Objectives

This work implements a coordination approach that improves the performance of decentralized control systems. This task begins by defining the objectives of the block controllers involved in the coordination. Recall that in this research each block has an MPC controller with an independent objective, which in the context of this project means that only the output variables of the block are controlled by using the input variables of the same block.

In order to illustrate the concept let's consider the formulation of the extended D-MPC controller described in Section 3.6 where the objective function of each controller may incorporate output variables from another blocks weighted by using parameter W_{ij}. The Block D-MPC controller considers the values of parameter W_{ij} to be zero because no importance is assigned to control output variables from other blocks.

It is important to point out that this algorithm does not intend to match the performance achieved by using a centralized MPC. In fact the goal is to obtain the good values for the individual block objective functions when each block controller adjusts only the manipulated variables within its own block. This goal is achieved with communication among blocks to reduce the negative affects of interaction.

5.2 Dynamic Performance and Case Studies

The goal of this research is to develop a Block MPC controller that provides autonomous control for each block with "good" dynamic behaviour for all variables. A direct comparison of performance among centralized MPC, independent block MPC and D-MPC would require all controllers to be tuned optimally. Thus, each tuning would have to provide the best performance as limited by a specific robustness guarantee. Solving this optimal tuning problem is a research project in itself, involving non-convex optimization. In the studies reported here, the tuning is performed by trial-and-error to provide reasonable transient responses. Therefore, the relative performances among various control structures should be interpreted as indicating whether a small or large difference exists between achievable performances among structures. However, these results demonstrate that D-MPC provides well-behaved transient responses that are similar to centralized control and that D-

MPC can stabilize some control systems that cannot be stabilized by independent block MPC.

While the processes considered are non-linear, the simulations cases use linearized dynamic models to represent the plant in the closed loop simulation studies. The following issues are general and are used in all the cases unless otherwise noted.

- The simulations studies have no mismatch between the model used by the controller and the model representing the plan.
- No noise has been added to the measurements.
- All controllers tuning has been performed by trial-and-error to provide reasonably fast responses of the controlled variables without undue variability (overshoot, oscillations, etc) in the manipulated variables.

5.2.1 Computational Requirements.

The computational requirements for the D-MPC controller are different for the constrained and the unconstrained cases. For example for the unconstrained D-MPC controller is important to note that the resulting system of linear equations is only of the following size $\Re^{\left(\sum_{i=1}^{N} m_i M_i\right) x \left(\sum_{i=1}^{N} m_i M_i\right)}$, where m_i and M_i are the input horizon and number of input variables in block i respectively. Furthermore, since the controller gain matrix, K_{DMPC} is known its calculation can be done in advance and only the corresponding rows of the K_{DMPC} matrix are required to compute the control action.

For the constrained D-MPC this work considers two different approaches.

(1) D-MPC that uses full set of KKT conditions: In this case the controller requires the solution of a system of nonlinear, non-convex equations. In this work this

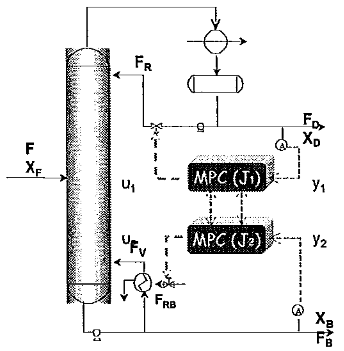
problem is solved using the solver IPOPT-C. The largest problem solved was a 3x3 with 150 variables and required an average CPU time of 0.032 sec per controller execution.

(2) Heuristic D-MPC: Since every active constraint adds an equation and a corresponding Lagrange multiplier to the unconstrained linear system this method requires solving a linear system of equations that grows one size per iteration. Therefore the K_{DMPC} matrix cannot be computed offline. Although the requirements of the heuristic D-MPC are larger than those of the unconstrained D-MPC the computations times remain so low that can be neglected, the largest problem solved required an average of 0.00015 CPU seconds per execution.

Finally, Appendix E shows a schematic description of the software used for the implementation of the D-MPC controller.

5.3 Case Study I: Binary Distillation Column.

The first case study considered for this project is a two-product, binary distillation column as described in Marlin (2000), the intention is to understand and analyze the performance and characteristics of the D-MPC controller. This is a multivariable system consisting of two inputs and two outputs for the composition control of a distillation column. Figure 5.1 presents a sketch of the control problem along with the variable description.



y_1 = Distillate mole fraction light key
y_2 = Bottoms mole fraction light key
u_1 = Reflux rate, kgmol/min
u_2 = Amount vaporized by reboiler, kgmol/min

Figure 5.1 Binary Distillation Column, (Positive BRG configuration)

The nominal, linearized model along with the disturbance model is given by.

$$\begin{pmatrix} X_D \\ X_B \end{pmatrix} = \begin{pmatrix} \dfrac{0.0747e^{-2.5s}}{12s+1} & \dfrac{-0.0667e^{-3.5s}}{15s+1} \\ \dfrac{0.1173e^{-3.0s}}{11.75s+1} & \dfrac{-0.1253e^{-2.3s}}{10.2s+1} \end{pmatrix} \begin{pmatrix} F_R \\ F_V \end{pmatrix} + \begin{pmatrix} \dfrac{0.07e^{-5s}}{14.4s+1} \\ \dfrac{1.3e^{-3s}}{12s+1} \end{pmatrix} (X_F) \tag{5.1}$$

In this case, two decentralized MPC controllers are used to control the column, the distillate composition is paired with the reflux flowrate $(X_D - F_R)$ and the bottoms composition is paired with the reboiled vapour $(X_B - F_V)$. The following RGA matrix suggests such pairing.

$$\Lambda = \begin{bmatrix} (6.094) & -5.094 \\ -5.094 & (6.094) \end{bmatrix} \tag{5.2}$$

The disturbance model represents the dynamic response of the process from changes in the feed composition. Reasonable tuning is used for the simulations, these parameters are assumed constant and are described in Table 5.1 unless other specified.

Table 5.1 Distillation Column Tuning Parameters

Parameter	Description	Value
m	Input horizon	*10*
p	Output horizon	*65*
Δt	Execution and sampling time, (min)	*2*
Q	Output variable weighting	$[1 \quad 1]$
R	Input Suppression Factor	$[0.2 \quad 0.2]$

5.3.1 Unconstrained D-MPC

First, let's consider the unconstrained D-MPC case where a system of linear equations is solved at every execution time to obtain the control actions. Since this is an unconstrained problem, so that the complementarity conditions are not present and the active set heuristic is not applied. The resulting controller is described in Section 3.2, equation (3.17).

In order to present a reference for the D-MPC performance the following plots show the simulation results of centralized MPC (C-MPC), D-MPC and independent block MPC controllers when subject to set point changes or disturbance changes. Each controller used the same tuning parameters during the simulations. Figure 5.2 shows a simulation for a set point change in X_D and in X_B. Both C-MPC and D-MPC controllers provide "well behaved responses, with little overshoot or oscillation and return to set points in a reasonable time. We can conclude that little performance in deviation from set point is lost by using the D-MPC in this case. The independent block MPC has the worst performance of the three controllers and presents a more sluggish dynamic.

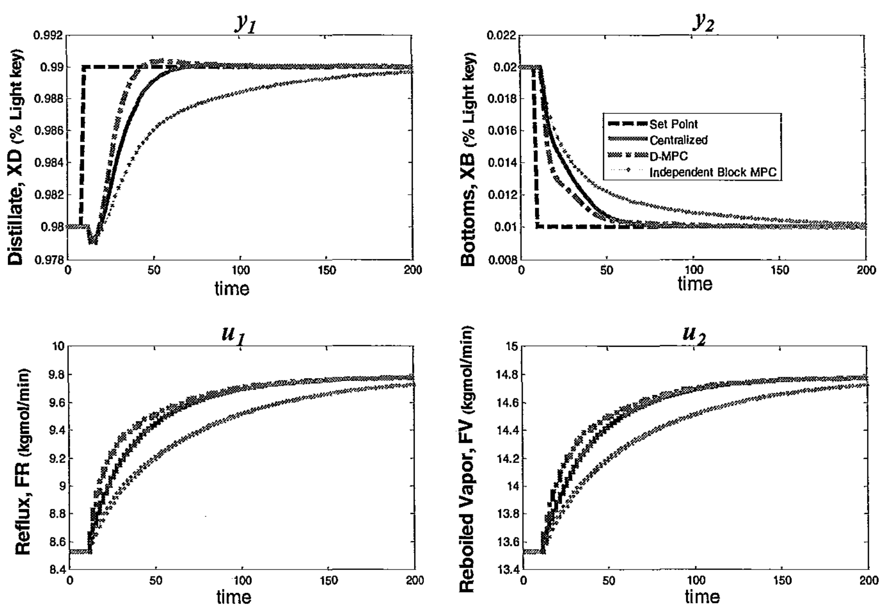


Figure 5.2 Dynamic Performance - Unconstrained D-MPC with Positive RGA

In order to compare the overall performance of the controllers Table 5.2 shows the integral of the squared error (ISE) as a performance measure. In this specific example the D-MPC presents a very good performance and the ISE is lower than that of the C-MPC controller. However, it is important to note that if we compare the performance of the input variables (i.e. the sum of squared movements, SSM) the performance of the D-MPC is not better, showing more aggressive adjustments of the manipulated variables. The SSM is calculated as the summation of all the squared input moves that were actually implemented in the process during the simulation. The behaviour obtained for this case is representative of most unconstrained cases.

Table 5.2 ISE and SSM, Distillation Column (Positive RGA)

	Variable	C-MPC	D-MPC	Independent Block MPC
ISE	$y_1(x_{TD})$	1.9453E-03	1.6338E-03	3.0259E-03
	$y_2(x_{SD})$	9.6852E-04	6.9976E-04	1.3709E-03
	Total	**2.9138E-03**	**2.3336E-03**	**4.3968E-03**
SSM	$u_1(F_R)$	2.8658E-02	4.4995E-02	1.5917E-02
	$u_2(F_V)$	2.5520E-02	3.6852E-02	1.5068E-02
	Total	**5.4178E-02**	**8.1847E-02**	**3.0985E-02**

5.3.2 Control Under Different Blockings (Negative and Zero RGA)

An important feature of a control design is integrity, which basically requires the sign of the process gain $\left(\Delta y / \Delta u\right)$ to be independent of the on/off status of the rest of the controllers. Integrity is not an absolute requirement for control design; we can accept designs without integrity for a substantial improvement in dynamic performance. A positive RGA or BRG is the minimum requirement for integrity in independent block MPCs involving SISO blocks. However, systems paired on variables with negative RGA elements can be controlled by this new D-MPC, while they cannot be stabilized with conventional independent block MPC technology. In addition, some coordination MPC methods cannot control negative RGA element pairings. As described in Section 4.2.3 Venkat and Rawlings (2004) analyzed a similar distillation column (Ogunnaike and Ray, 1994) considering a negative RGA configuration. They found that no communication approach (achieving Nash Equilibrium) is able to stabilize the plant. Additionally Li et al (2005) developed an iterative Communication-MPC approach where the successful convergence of the method was limited to with a diagonal dominant matrix. Here, the performance of D-MPC is demonstrated for cases with negative and zero RGA configurations. The D-MPC obtained successful results and was able to provide integrity for the system. The following case shows that the D-MPC is able to converge and stabilize the system even in configurations with negative RGA (BRG). For this example we intentionally consider pairing on a negative configuration where the interactions are dominant, resulting in the following pairings; $(X_B - F_R)$ and $(X_D - F_V)$.

The dynamic simulation is illustrated in Figure 5.3 it compares the performance of the D-MPC controller with the benchmark C-MPC controller and the independent block MPC technology.

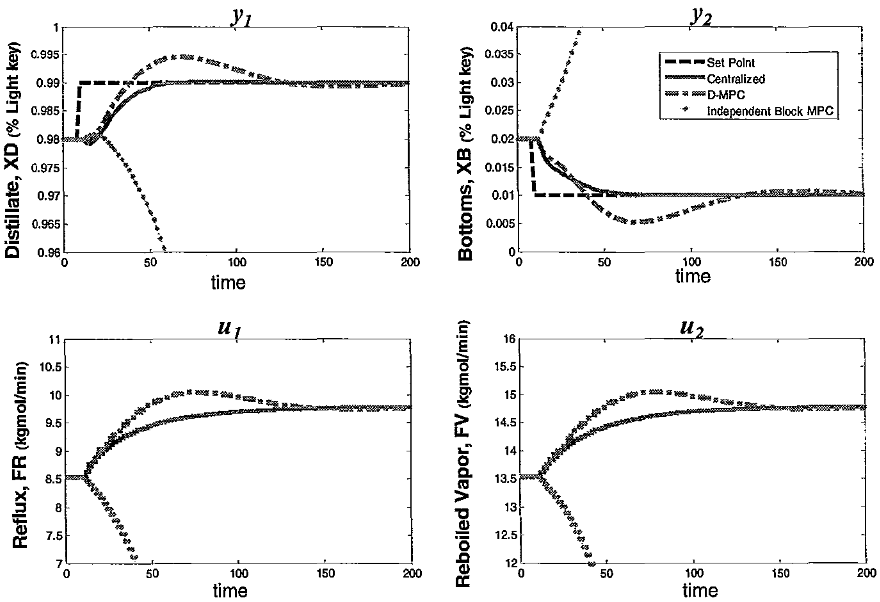


Figure 5.3 Dynamic Performance - Unconstrained D-MPC with Negative RGA

In this simulation the D-MPC controller is able to provide set point tracking while the independent block MPC is not even able to stabilize the plant. Without a doubt this is a big improvement to the current practice and one of the main advantages of the novel D-MPC controller over the iterative algorithms presented in literature. Table 5.3 shows the ISE for the three different controllers.

Table 5.3 ISE, Distillation Column (Negative RGA)

	Variable	C-MPC	D-MPC	Independent Block MPC
ISE	$y_1(x_{TD})$	1.9453E-03	2.3917E-03	∞
	$y_2(x_{SD})$	9.6852E-04	2.0303E-03	∞
	Total	**2.9138E-03**	**4.4220E-03**	∞

To further analyze the previous case study, we consider the convergence criteria of Li et al. (2005), which requires that the spectral radius of the following expression must be less than one to guarantee a convergent computation $|\rho(D_0)| < 1$. Where D_0 is computed as following.

$$D_0 = \begin{pmatrix} 0 & \left(A_{11}^T Q_1 A_{11} + R_1\right)^{-1} A_{11}^T Q_1 A_{12} \\ \left(A_{22}^T Q_2 A_{22} + R_2\right)^{-1} A_{22}^T Q_2 A_{21} & 0 \end{pmatrix}$$

Using the tuning parameters described in Table 5.1 the spectral radius calculated for this example is $|\rho(D_0)| = 1.141$ which is in clearly violation of the convergence criteria. According to Li et al (2005) and Venkat and Rawlings (2004) iterative Communication-MPC approaches are not able to achieve control of this column.

In contrast, the stability analysis method described in Section 4.2.2 validates the performance of the D-MPC controller. Figure 5.4 shows that the D-MPC controller has all the eigenvalues of the closed loop expression inside the unitary circle while the independent block MPC has at least one eigenvalue outside the circle.

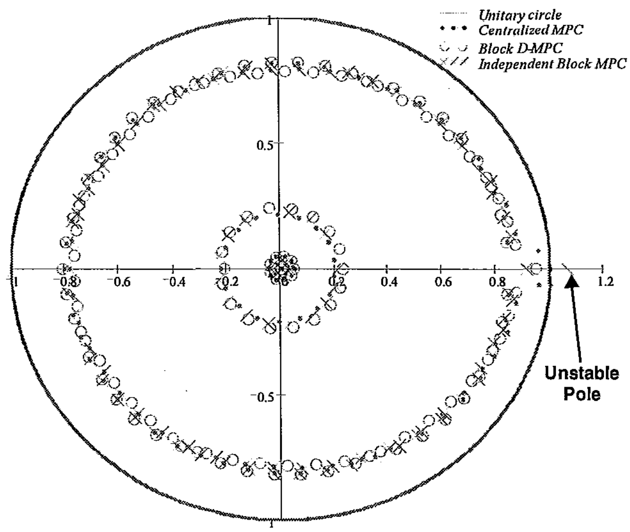


Figure 5.4 Nominal Stability - Unconstrained D-MPC with Negative RGA

It is important to mention that while some tuning combinations for the D-MPC controller may result in unstable behaviour the simulation in Figure 5.3 and results in Figure 5.4 proved that there exist tuning combinations that allows for stable behaviour. A more extensive analysis of the stability region for this negative RGA configuration is presented in Appendix C.

Another important case considered in this research is the case of **zero RGA** configurations. In these cases, the process has a zero gain matrix implying that there is no causal relation in one of the blocks (i.e. $y_2 - u_2$) and the causal relation is only through interaction processes. In order to overcome this obstacle a relatively small model mismatch is introduced in the D-MPC controller. The distillation column is now modified to illustrate procedure; basically one of the process gains (Kp_{22}) is set to zero. It is important to mention that this is a hypothetical problem in order to test the capabilities of the D-MPC controller.

(Process Gain Matrix)	(Model Gain Matrix)
$Kp = \begin{pmatrix} 0.0747 & -0.0667 \\ 0.1173 & 0 \end{pmatrix}$	$Kp_m = \begin{pmatrix} 0.0747 & -0.0667 \\ 0.1173 & \varepsilon \end{pmatrix}$

In order to solve the zero-gain problem parameter $\varepsilon = 0.001$ is introduced in the model used in the controller. Here it is important to mention the reason why this mismatch, ε is required. For this specific case study if $\varepsilon = 0$ then $A_{22} = 0$ and the D-MPC controller will only produce arbitrary solutions determined merely by the move suppression factor of the second controller, R_2. Furthermore if the move suppression is zero $(R_2 = 0)$ the D-MPC controller becomes singular. This can be easily illustrated by expanding the D-MPC controller for this *two by two* case with $\varepsilon = 0$ (For full controller, see Section 3.5 equation (3.40))

$$\begin{pmatrix} \Delta u_1 \\ \Delta u_2 \end{pmatrix} = \begin{pmatrix} A_{11}^T Q_1 A_{11} + R_1 & A_{11}^T Q_1 A_{12} \\ 0 & 0 + R_2 \end{pmatrix}^{-1} \begin{pmatrix} A_{11}^T Q_1 (-e_1) \\ 0 \end{pmatrix}$$

Let us note that the magnitude of ε was obtained by trial-and-error and using the stability analysis. In this way the sign of ε is not important as long as the magnitude is sufficiently small. Figure 5.5 shows that the performance of the D-MPC controller for a zero-gain system can be similar to that of the centralized controller.

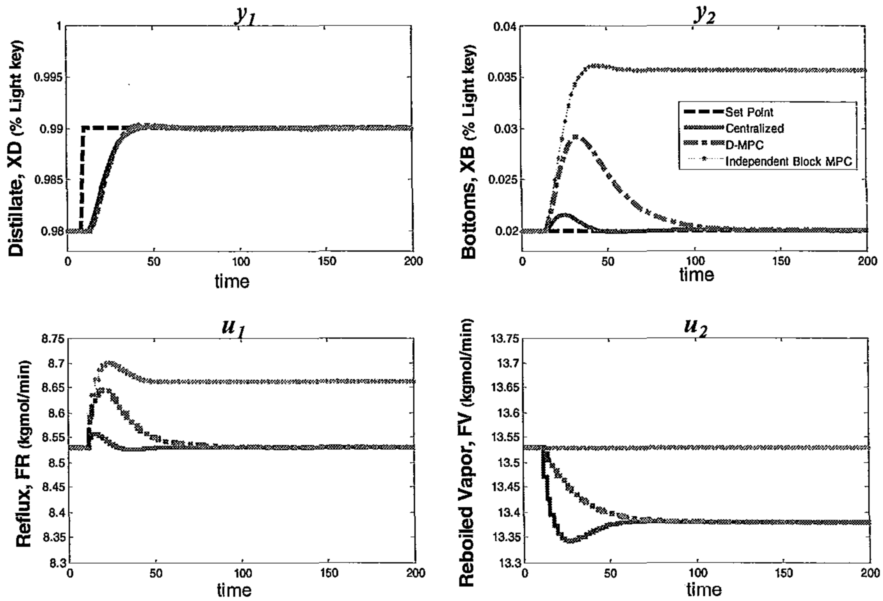


Figure 5.5 Dynamic Performance - Unconstrained D-MPC with Zero RGA

This is an interesting behaviour because the D-MPC controller was able to control both variables with a relatively small degradation of performance as illustrated in Table 5.4. Again it is important to mention that other published technology cannot control systems with a zero-RGA configuration. The D-MPC on the other hand and by means of a simple strategy is able to control the system.

Table 5.4 ISE, Distillation Column (Zero RGA)

	Variable	C-MPC	D-MPC	Independent Block MPC
ISE	$y_1(x_{TD})$	1.0100E-03	1.1071E-03	1.1407E-03
	$y_2(x_{SD})$	2.8692E-05	2.1452E-03	∞ *
	Total	**1.0387E-03**	**3.2523E-03**	∞

* ISE values are infinity for offset results.

5.3.3 Constrained D-MPC

One of the main advantages of any MPC application is the capability to handle constraints. Basically the set of linear equations is now extended into a set of nonlinear equations that include complementarity equations, which can be replaced through the active set heuristic. However, it is important to note that the incorporation of input and output constraints and the presence of active constraints (saturation) may lead to loss of degrees of freedom that results in steady-state offset in the system. In the following examples the input bounds are adjusted accordingly.

One of the main goals of the proposed D-MPC approach is clearly illustrated in the following simulation, which shows the D-MPC controller isolating a saturation effect within one block. This case is simulated using both D-MPC proposed methods, (1) the KKT approach with an IPOPT-C solver described in Section 3.3 and (2) the heuristic D-MPC described in Section 3.4. From Figure 5.6 it can be observed that with D-MPC, a steady state offset occurs only in the subsystem with input saturation, $(X_D - F_R)$ while the other subsystem, $(X_B - F_V)$ returns its controlled variable to its set point by using its free input variable.

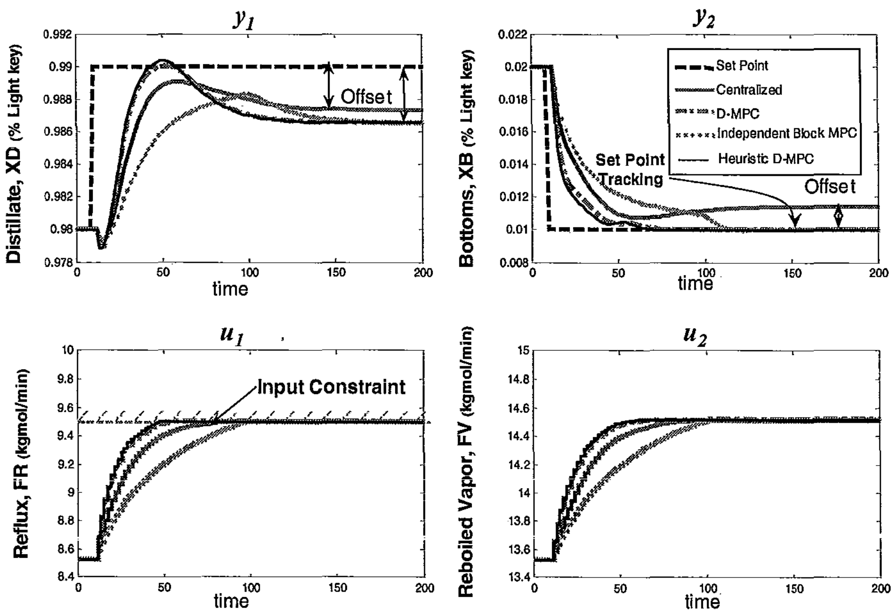


Figure 5.6 Dynamic Performance - Constrained D-MPC with Positive RGA

In contrast, the saturation reached in the reflux flow (F_R) precludes the C-MPC controller from returning both controlled variables to their set points. As a result, the C-MPC controller distributes the steady-state errors between both controlled variables, therefore neither controlled variable returns to the set point. The set points and final steady states of the C-MPC and both D-MPC controllers are described in Table 5.5. The performance of D-MPC (ISE) with first-order KKT conditions and the heuristic D-MPC are practically the same with about a *5 %* difference. This example shows how the D-MPC controller respects the local autonomy of each subsystem by not allowing the saturation problem in the one loop to affect the set point tracking of the second loop.

Table 5.5 ISE and Final Steady States, Distillation Column (Positive RGA)

Variable	C-MPC		D-MPC		Heuristic D-MPC		Independent Block MPC	
	Steady State	ISE	Steady State	ISE	Steady State	ISE	Steady State	ISE
$X_D\,(0.99)$	**0.9873****	∞	**0.9866****	∞	**0.9866****	∞	**0.987****	∞
$X_F\,(0.01)$	**0.0114****	∞	**0.010 ***	6.73E-04	**0.010 ***	6.39E-04	**0.010 ***	13.4E-04
F_R	9.50		9.50		9.50		9.50	
F_V	14.51		14.52		14.52		14.52	

** Offset, * Set point tracking,

The computational requirements although larger than the unconstrained problem can be neglected for practical reasons. For this problem IPOPT-C reported an average of 0.02 CPU seconds for each D-MPC execution.

The D-MPC controller has shown to be effective when solving strongly interactive systems with positive RGA configurations. To further test the capabilities of this method constrained cases were also tested for negative and zero RGA configurations. However, on such cases the D-MPC controller was not able to solve the problem when the system encountered saturation. The solver (IPOPT-C) found a local solution that was obviously incorrect, e.g., it did not return controlled variables to set point when it was possible.

In order to achieve a reliable solution the active set heuristic methodology described in Section 3.4 was successfully employed. A test case with a negative RGA configuration is now presented. For this case the heuristic strategy requires the following linear system of equations to be solved in an iterative way at every controller execution.

$$\begin{pmatrix} \Delta u_1 \\ \Delta u_2 \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} A_{21}{}^T Q_2 A_{21} + R_1 & A_{21}{}^T Q_2 A_{22} & 0 & H_1{}^T \\ A_{12}{}^T Q_1 A_{11} & A_{12}{}^T Q_1 A_{12} + R_2 & H_2{}^T & 0 \\ 0 & H_2 & 0 & 0 \\ H_1 & 0 & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} A_{21}{}^T Q_2(-e_2) \\ A_{12}{}^T Q_1(-e_1) \\ B_1 \\ B_2 \end{pmatrix} \tag{5.3}$$

$$H_i = \begin{bmatrix} 1 & 0 & 0 \\ \vdots & \ddots & 0 \\ 1 & \cdots & 1 \end{bmatrix} \tag{5.4}$$

Where H_i are the time variant constraints that contain the information on the set of active constraints. Figure 5.7 shows the closed loop simulation of the distillation plant subject to a set point change on $y_1(X_D)$. The reboiled vapour, u_2 which is now controlling $y_1(X_D)$ gets saturated leading to an offset in this controlled variable. The other loop however maintains a satisfactory set point tracking. The set points and final steady states of both C-MPC and D-MPC controllers are reported in Table 5.6 along with the transient performance indexes.

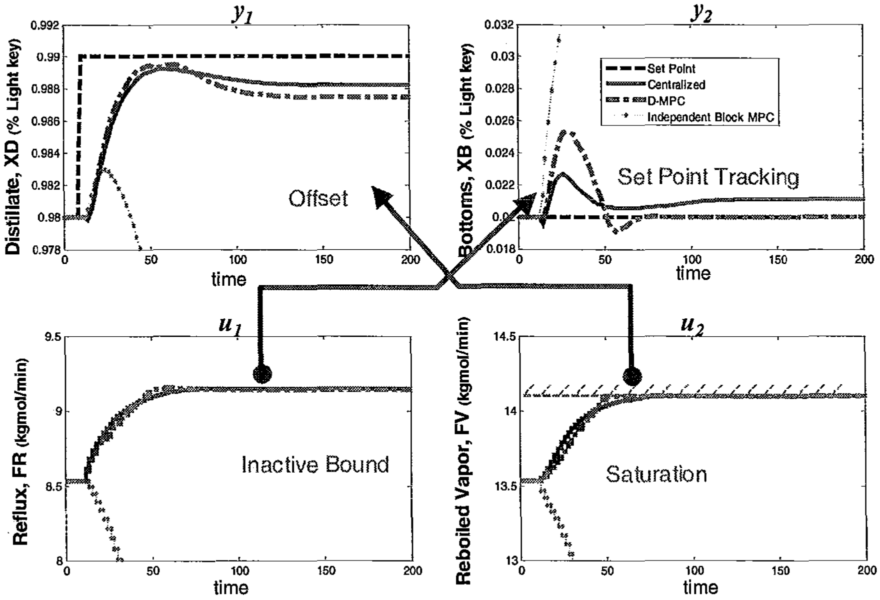


Figure 5.7 Dynamic Performance – Constrained D-MPC with Negative RGA

Table 5.6 ISE and Final Steady States, Distillation Column (Negative RGA)

Variable (Set Point)	C-MPC		D-MPC	
	Steady State	ISE	Steady State	ISE
$X_D\,(0.99)$	**0.988****	∞	**0.987****	∞
$X_F\,(0.02)$	**0.02115****	∞	**0.02***	0.000498
F_R	9.15		9.14	
F_V	14.1		14.1	

** Offset, * Set point tracking.

This example shows the advantages of the heuristic strategy over the D-MPC that uses the full KKT conditions. Let's recall some of the main characteristics that support this heuristic method.

(1) The best control solution is not guaranteed but a feasible solution is always achieved. If the correct active set is reached, the heuristic will recognize this as a solution.

(2) Finite and low number of iterations.

(3) Computational requirements are reduced to the sequential solution of a set of linear equations.

(4) Good computation experience for cases with different integrity (positive, negative and zero RGA)

(5) Years of experience in industry with similar method applied to C-MPC (Prett et al. 1980; Richalet et al 1987)

Finally, it is important to remark that all simulations where performed considering no model-mismatch or measurement noise in order to clearly illustrate the performance of the controllers. Appendix D shows a couple D-MPC examples that evaluate the dynamic performance when model-mismatch is present.

5.4 Case Study II: Shell Standard Control Problem.

We now apply the D-MPC controller to an industrial benchmark problem known as the Shell Standard Control Problem first proposed by Prett and Morari, (1987). The process is illustrated in Figure 5.8 it consists of a heavy oil fractionator with three product draws and three side circulating loops. The three circulating loops remove heat at high temperatures, rather than at the lower temperature of the condenser; they are called pumparounds. The pumparound heat exchangers in the top two circulating loops, which are mid-tower condensers, are used as reboilers for other columns; their duties act as disturbances to the column.

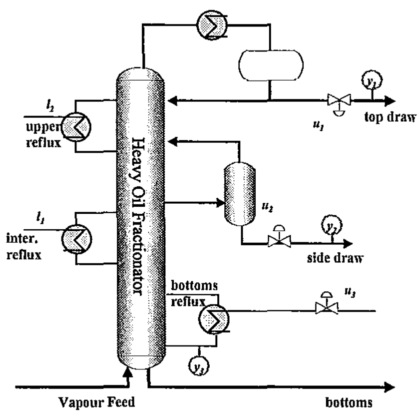


Figure 5.8 Schematic of the Shell Heavy Oil Fractionator.

The dynamic model is given in the following, without engineering units to be consistent with the original citation.

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \frac{4.05e^{-27s}}{50s+1} & \frac{1.77e^{-28s}}{60s+1} & \frac{5.88e^{-27s}}{50s+1} \\ \frac{5.39e^{-18s}}{50s+1} & \frac{5.72e^{-14s}}{60s+1} & \frac{6.9e^{-15s}}{40s+1} \\ \frac{4.38e^{-20s}}{33s+1} & \frac{4.42e^{-22s}}{44s+1} & \frac{7.20}{19s+1} \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \begin{pmatrix} \frac{1.20e^{-27s}}{45s+1} & \frac{1.44e^{-27s}}{40s+1} \\ \frac{1.52e^{-15s}}{25s+1} & \frac{1.83e^{-15s}}{25s+1} \\ \frac{1.14}{27s+1} & \frac{1.26}{32s+1} \end{pmatrix} \cdot \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \tag{5.5}$$

Where y_1 is the top draw composition (x_{TD}), y_2 is the side draw composition (x_{SD}) and y_3 is the bottoms reflux temperature (T_{BR}). Manipulated inputs are the top (F_{TD}) and side (F_{SD}) draws which are u_1 and u_2 respectively, and u_3 is the bottoms reflux duty F_{BR}. The disturbances l_1 and l_2 are the heat duties from the top circulating loops.

The RGA for this process is shown in (5.6), it suggest a diagonal control structure.

$$\Lambda = \begin{pmatrix} 2.08 & -0.73 & -0.35 \\ 3.42 & 0.93 & -3.36 \\ -4.5 & 0.79 & 4.71 \end{pmatrix} . \tag{5.6}$$

The **control objectives** considered for this case are pretty much the same as in the original problem (Prett and Morari, 1987), except that we removed the control of u_3 and added the control of y_3. The resulting control objectives are the following.

(1) Maintain the output variables y_1, y_2 and y_3 at specified set points (0.0 with tolerance of 0.005 at steady-state.)

(2) Reject disturbances l_1 and l_2 entering the columns.

The control constraints are.

(1) All control inputs must be maintained at: $|u_f| \leq 0.5$ $(f = 1,2,3)$ unless other specified

(2) Maximum input size control of $|\Delta u_f| \leq 0.2$

A challenge problem considering the original control objectives will be addressed in Section 5.4.4.

5.4.1 Block Control Structure

Obtaining the best control structure for a specific problem is a very challenging problem because the possible number of block structures is greater than the multiloop structure and grows exponentially with the system dimension (Cai, 2009) because the block sizes and variables allocation to blocks must be decided. The structure design problem is beyond the scope of this work; therefore, only a couple of structures are considered to perform simulations. Figure 5.9 shows a scheme of some of the possible block structures for this three by three process, it ranges from the multiloop structure, (S.1) to the centralized structure, (S.5). The other structures (S.2-S.4) involve two blocks.

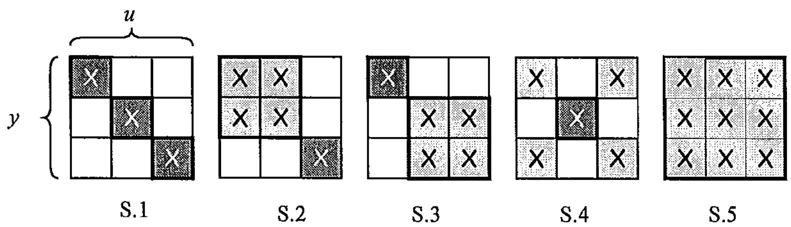


Figure 5.9 Block Structures for the Shell Heavy Oil Fractionator

The block relative gain for structures S.2, S.3 and S.4 are presented in Table 5.7.

Table 5.7 Block Relative Gain

Control Structure	BRG*
(S.2) - $[y_1 y_2 - u_1 u_2],[y_3 - u_3]$	$\lvert \Lambda_{B11} \rvert = 4.71$
(S.3) - $[y_1 - u_1],[y_2 y_3 - u_2 u_3]$	$\lvert \Lambda_{B11} \rvert = 2.08$
(S.4) - $[y_2 - u_2],[y_1 y_3 - u_1 u_3]$	$\lvert \Lambda_{B11} \rvert = 0.934$

$^*\lvert \Lambda_{B11} \rvert = \lvert \Lambda_{B22} \rvert$

All of these structures including S.1 have a positive BRG. However, this work will consider structure S.2 as the most challenging candidate for the simulations since it has the largest block relative gain of the three two-block control structures. The first block of structure S.2 consists of a multivariable composition controller $[x_{TD} x_{SD} - F_{TD} F_{SD}]$ while the second block consists of a temperature controller $[T_{BR} - F_{BR}]$. Figure 5.10 shows a scheme of the control structure for this case.

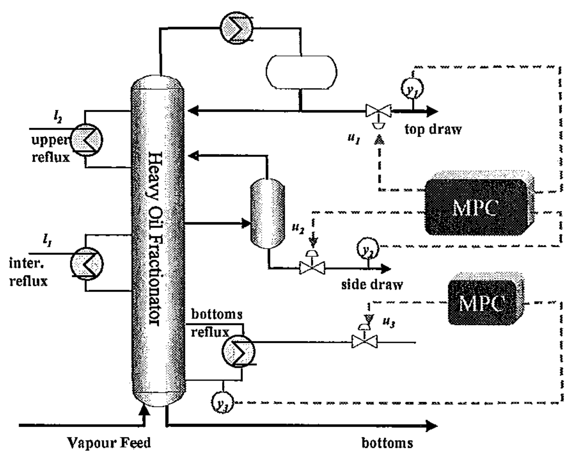


Figure 5.10 Block Structure S.2 (Positive BRG)

5.4.2 Block D-MPC Controller

The unconstrained D-MPC controller required for this block structure (S.2) is the following.

$$\Delta u = \left(A_D{}^T Q\ A_C + R\right)^{-1} A_D{}^T Q \cdot (-e) \tag{5.7}$$

Where *Block 1* is a *two by two* MPC controller with the following variables $[y_1 y_2 - u_1 u_2]$ and *Block 2* is a SISO MPC controller for $[y_3 - u_3]$. In order to implement this control structure, matrices A_C and A_D are defined as follows.

$$A_C = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) \quad A_D = \left(\begin{array}{c|c} A_{11} & 0 \\ \hline 0 & A_{22} \end{array}\right) \tag{5.8}$$

Where $A_{ij} \in \Re^{(P_i p \, x M_j m)}$ $(i, j = 1,2)$ contains either a SISO or a MIMO dynamic matrix that describes the effect of inputs in block j to outputs in block i. Moreover A_C can be defined as follows.

$$A_C = \left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) = \left(\begin{array}{cc|c} \hat{A}_{11} & \hat{A}_{12} & \hat{A}_{13} \\ \hat{A}_{21} & \hat{A}_{22} & \hat{A}_{23} \\ \hline \hat{A}_{31} & \hat{A}_{32} & \hat{A}_{33} \end{array}\right) \tag{5.9}$$

Where $\hat{A}_{gf} \in \Re^{(p x m)}$ are the dynamic matrices relating the changes in output g due to changes in input f. In the same way A_D can be defined as follows.

$$A_D = \left(\begin{array}{cc|c} \hat{A}_{11} & \hat{A}_{12} & 0 \\ \hat{A}_{21} & \hat{A}_{22} & 0 \\ \hline 0 & 0 & \hat{A}_{33} \end{array}\right) \tag{5.10}$$

The tuning parameters $Q_i \in \Re^{(P_i p \, x P_i p)}$ and $R_i \in \Re^{(M_i m \, x M_i m)}$ $(i = 1,2)$ are diagonal matrices. Table 5.8 shows the tuning parameters to be used in the following simulations, they are kept constant unless other specified.

Table 5.8 D-MPC tuning parameters for the Shell Oil Fractionator

Parameter	Description	Value
m	Input horizon	5
p	Output horizon	80
Δt	Execution and sampling time	5
Q	Output variable weighting	$[1 \;\; 1 \;\; 1]^T$
R	Input suppression factor	$[10 \;\; 10 \;\; 10]^T$

Figure 5.11 shows the dynamic response of the selected control structure when subject to the disturbance changes, $l_1 = -0.5, l_2 = 0.5$. The response is plotted against independent block MPC and centralized MPC.

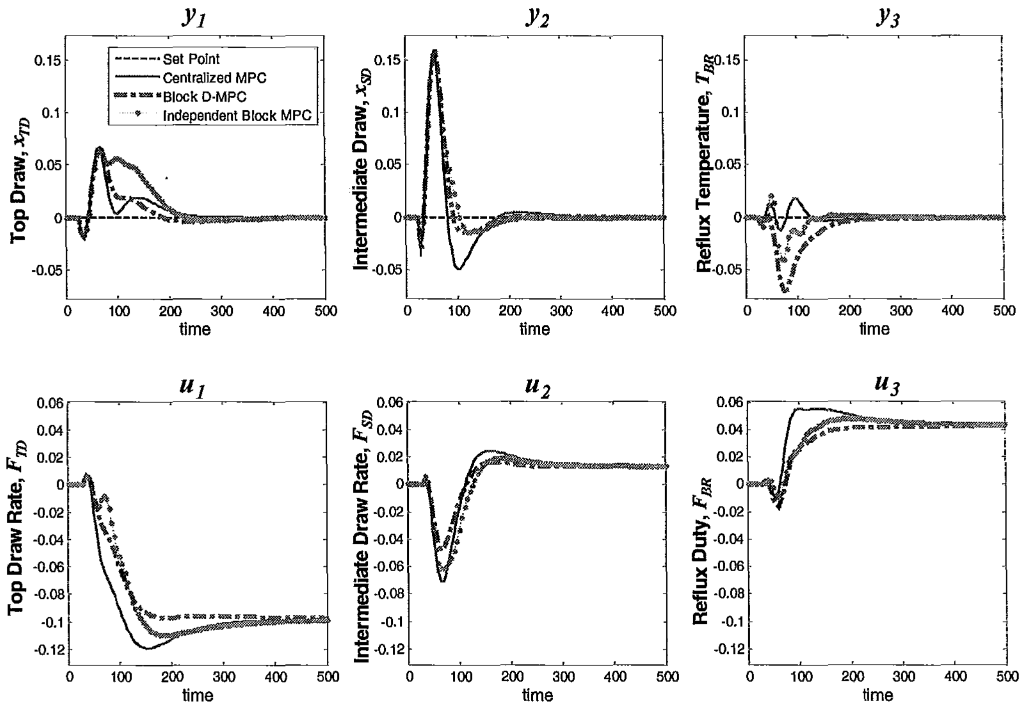


Figure 5.11 Dynamic Performance – Unconstrained D-MPC with Positive BRG

The dynamic performance is relatively good with a set point tracking of all three controlled variables at steady state. Table 5.9 shows the integral of the squared error for the three different controllers. It can be observed that the performance of the D-MPC controller is between the centralized MPC and the independent block MPC. There appears to be no significant difference in performances.

Table 5.9 Integral of the Squared Error *[l_1=-0.5, l_2=0.5]*

	Variable	C-MPC	D-MPC	Independent Block MPC
ISE	$y_1(x_{TD})$	0.1199	0.1371	0.3045
	$y_2(x_{SD})$	0.6151	0.6063	0.6273
	$y_3(T_{BR})$	0.0106	0.1707	0.0357
	Total	**0.746**	**0.914**	**0.967**

In order to test the constrained D-MPC controller a small modification is made to the manipulated variable bounds. Let's consider that the capacity of the bottoms reflux duty is reduced to $|u_3| \leq 0.02$. Due to the limited capacity of the bottoms reflux duty, it is not possible to completely reject both disturbances ($l_1 = -0.5, l_2 = 0.5$); therefore, offsets are to be expected. Figure 5.12 shows the dynamic performance subject to the same set of disturbance changes, and Table 5.10 shows the final steady states achieved by both controllers.

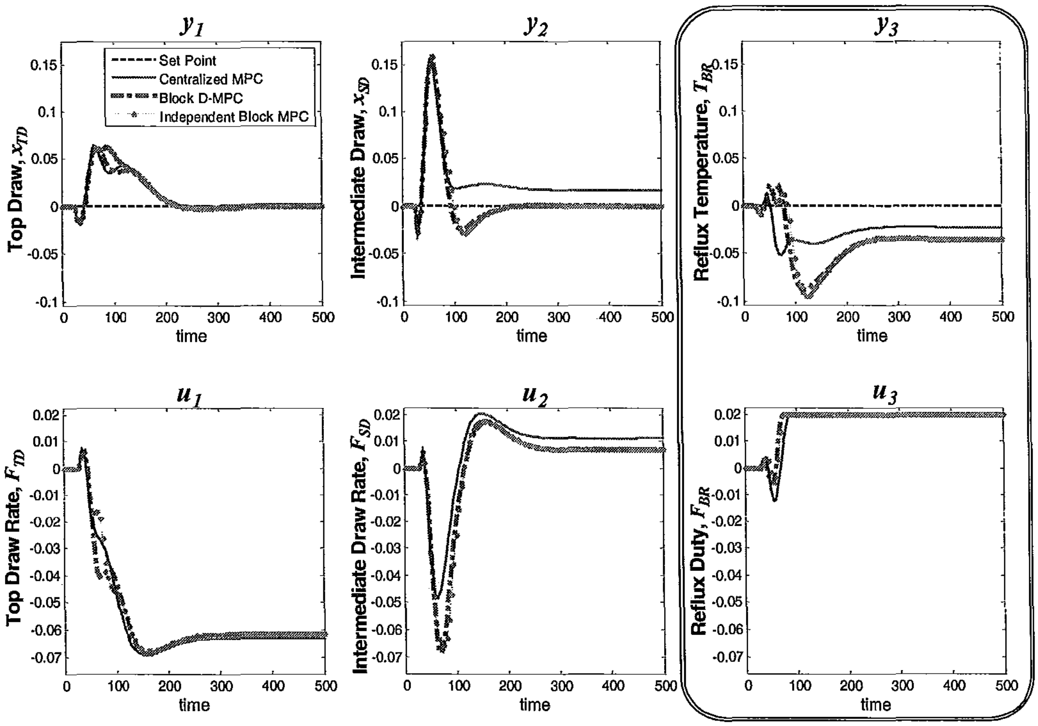


Figure 5.12 Dynamic Performance - Constrained D-MPC with Positive BRG

Table 5.10 Final Steady States [l_1=-0.5, l_2=0.5]

Variable	C-MPC		D-MPC	
	Steady State	ISE	Steady State	ISE
$y_1(x_{TD})$	0.002 **	∞	0.000 *	0.217
$y_2(x_{SD})$	0.017 **	∞	0.000 *	0.631
$y_3(T_{BR})$	- 0.023 **	∞	- 0.036 **	∞
$u_1(F_{TD})$	-0.063		-0.06166	
$u_2(F_{SD})$	0.011		0.007	
$u_3(F_{BR})$	0.02		0.02	

** Offset, * Set point tracking,

The D-MPC controller enforces the local autonomy of both blocks, it basically isolates the input saturation occurred in the temperature controller (*Block 1*) allowing the composition controller (*Block 2*) to successfully maintain output variables y_1 and y_2 at their respective set points at steady state. The average CPU time required at each D-MPC execution was of 0.032 seconds. The centralized MPC controller on the other hand results in offsets to all three outputs; two of them even violate the maximum tolerance allowed.

5.4.3 Zero and Negative BRG Configurations.

We continue with the Shell Challenge Problem in this section. The first case in this section considers a zero BRG control structure, which it is not usually employed in practice in part due to the fact that conventional technology is not able to control the process. In order to test this case a hypothetical version of the Shell problem is considered. Basically the steady-state gain matrices of the process (Kp) and model (Kp_m) are modified as follows.

$$Kp = \begin{pmatrix} 4.05 & 1.77 & 5.88 \\ 5.39 & 5.72 & 6.9 \\ 4.38 & 4.42 & 0 \end{pmatrix} \qquad Kp_m = \begin{pmatrix} 4.05 & 1.77 & 5.88 \\ 5.39 & 5.72 & 6.9 \\ 4.38 & 4.42 & \varepsilon \end{pmatrix} \tag{5.11}$$

Where $\varepsilon = 0.01$ introduces a small model mismatch sufficient for the D-MPC to be able to control the entire plant. The block structure remains the same with *Block 1* as the multivariable composition controller and *Block 2* as the temperature controller. The dynamic simulation is illustrated in Figure 5.13 and performance parameters are shown in Table 5.11, the following can be observed.

(1) The independent block MPC controller is by no means able to provide control of *Block 2*.

(2) The D-MPC uses the interaction models embedded in the controller to provide zero offset for y_3 in *Block 2*. However, it is important to mention that the

dynamic performance of *Block 2* suffers from the lack of direct causal effect between y_3 and u_3.

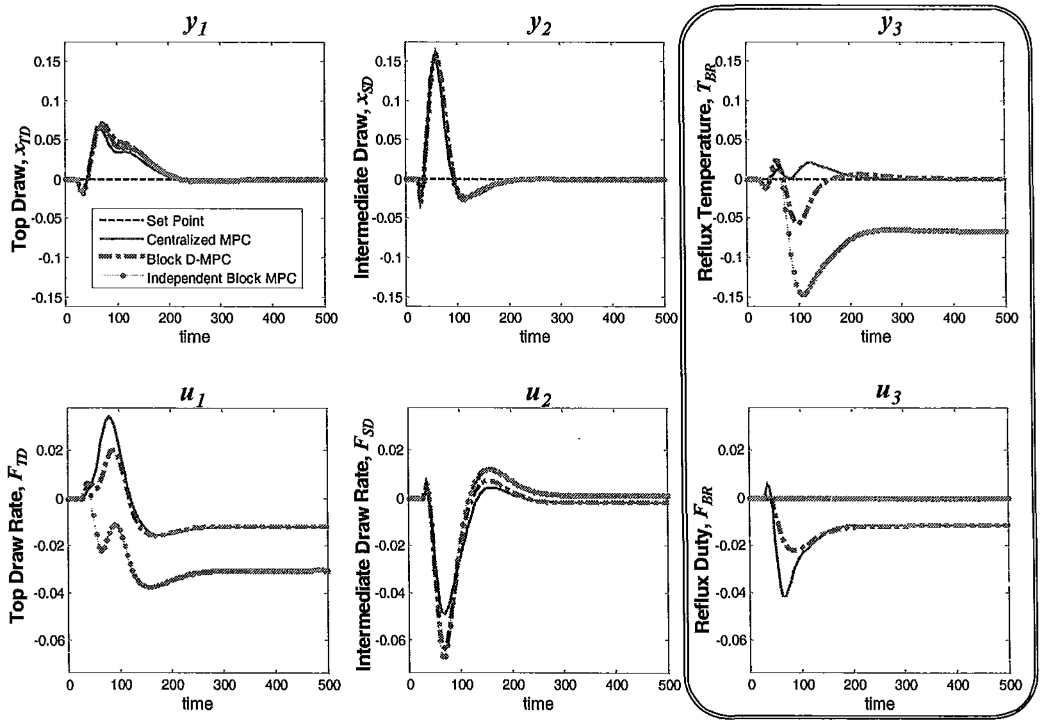


Figure 5.13 Dynamic Performance with Zero BRG

Table 5.11 Integral of the Squared Error [l_1=-0.5, l_2=0.5] - Zero BRG

	Variable	C-MPC	D-MPC	Independent Block MPC
ISE	$y_1(x_{TD})$	0.0400	0.0571	0.0509
	$y_2(x_{SD})$	0.1208	0.1336	0.1319
	$y_3(T_{BR})$	0.0049	0.0229	∞
	Total	**0.1657**	**0.2137**	∞

A more challenging test case where the interactions are so large that they become dominant is now addressed. It considers a less conventional block structure with a negative block relative gain $\left(|\Lambda_{B_{11}}| = -4.5\right)$. Figure 5.14 shows a schematic of the proposed control structure (S.6). We note that both blocks keep the same structure regarding the output variables, and only the input variables are redirected to different blocks.

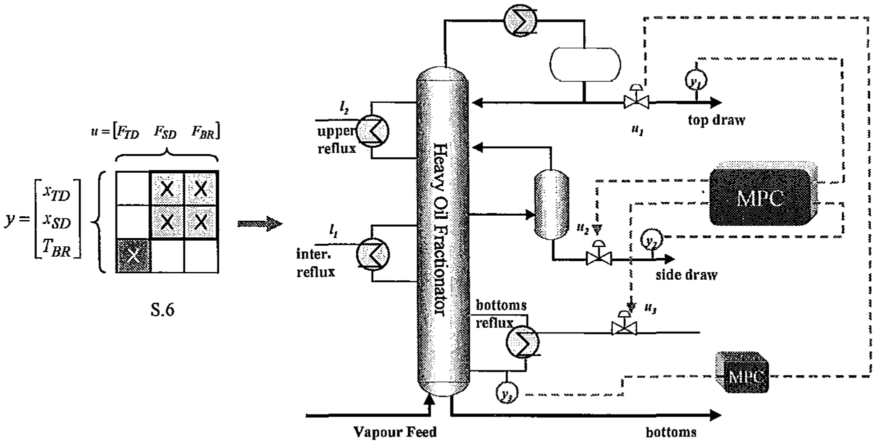


Figure 5.14 Block Structure S.6 – Negative BRG.

Figure 5.15 illustrates the stability analysis of both the D-MPC controller and the independent block MPC controller. It can be observed that a pole of the independent block MPC controller lies outside the unit circle, which indicates closed loop unstability.

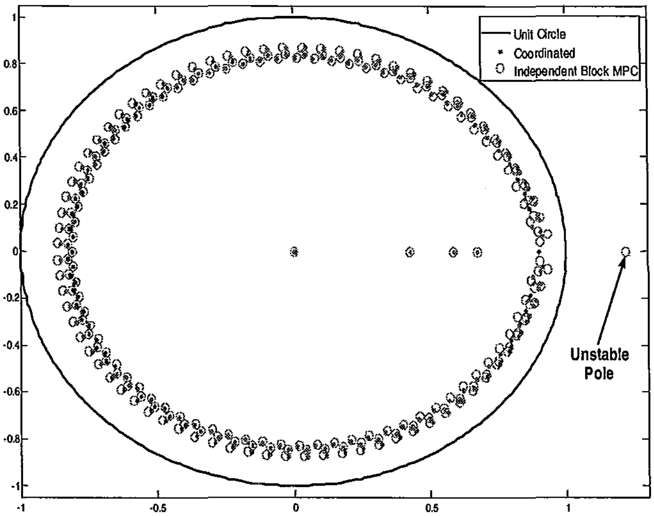


Figure 5.15 Nominal Stability (Negative BRG)

Figure 5.16 illustrates the dynamic performance of this negative BRG control structure. As predicted in the stability analysis the independent block MPC controller becomes unstable. The D-MPC however is able to achieve a zero steady state offset. The corresponding performance indexes are shown in Table 5.12.

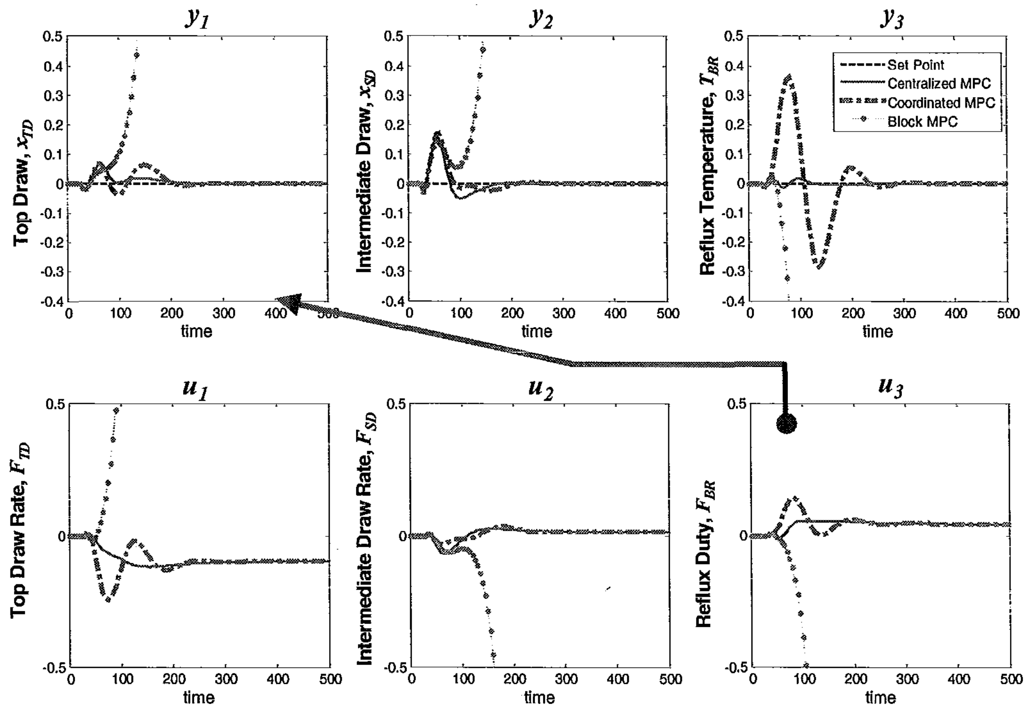


Figure 5.16 Dynamic Performance with Negative BRG

Table 5.12 Integral of the Squared Error [l_1=-0.5, l_2=0.5] – Negative BRG

	Variable	C-MPC	D-MPC	Independent Block MPC
ISE	$y_1(x_{TD})$	0.0240	0.0548	∞
	$y_2(x_{SD})$	0.1230	0.1383	∞
	$y_3(T_{BR})$	0.0021	1.3273	∞
	Total	0.1491	1.5204	∞

From this section it is important to remark that designs that under independent block MPC are unstable are able to provide a satisfactory performance with D-MPC. D-MPC performance is not much worse than that of a centralized MPC controller. These results demonstrate that D-MPC extends the range of processes and block designs for which block MPC is possible.

5.4.4 Challenge Problem: Alternative Control Objectives.

The original Shell Standard Control Problem considers different **control objectives** to those described in the previous section. The differences are the following.

(3) Maintain the input variable u_3 as close to -0.5 as possible (This maximizes the steam made in the bottoms reflux condenser.)

(4) Delete the control of output variable y_3.

The change requires a modification in the objective function of the MPC controller. Basically a third term is added to minimize the difference between the input variable and its reference value.

$$\underset{\Delta u_i}{Min}\ J_i = \frac{1}{2}\cdot\left(\left\| y_i - y_i^{sp}\right\|_{Q_i}^2 + \left\|\Delta u_i\right\|_{R_i}^2 + \left\| u_i - u_i^{ref}\right\|_{\alpha_i}^2\right) \tag{5.12}$$

The treatment of the controller is practically the same as described in Section 3.1.1. A brief development of this third term is now presented. First the input variable u_i must be put in terms of Δu_i.

$$u_i = u_{i_{(k)}} + \widetilde{V}_i \cdot \Delta u_i \tag{5.13}$$

Where $u_{i_{(k)}}$ is the current value of u_i and $\widetilde{V}_i \in \Re^{M_i m_i \times M_i m_i}$ is a matrix built by M_i diagonal blocks of lower triangular matrices of ones, ($V \in \Re^{m_i \times m_i}$ as described in Section 3.3.1, equation (3.21)). We can expand the third term in the objective function as follows.

$$\begin{aligned}\left\| u_i - u_i^{ref}\right\|_{\alpha_i}^2 &= \left\| u_{i(k)} + \widetilde{V}_i \cdot \Delta u_i - u_i^{ref}\right\|_{\alpha_i}^2 \\ &= \left\|\widetilde{V}_i \cdot \Delta u_i + \eta_i\right\|_{\alpha_i}^2 \\ &= \Delta u_i^T\left(\widetilde{V}_i^T \alpha_i \widetilde{V}_i\right)\Delta u_i + 2\eta_i^T \widetilde{V}_i \Delta u_i\end{aligned}$$

Where

$$\eta_i = \left(u_{i(k)} - u_i^{ref}\right)$$

Finally substituting this term in the objective function, J_i we get.

$$\underset{\Delta u_i}{Min}\ J_i = \frac{1}{2}\Delta u_i^T\left(A_{ii}^T Q_i A_{ii} + \underbrace{\tilde{V}^T \alpha_i \tilde{V}} + R_i \right)\Delta u_i + \left[A_{ii}^T Q_i (e_i + E_i) + \underbrace{\eta_i^T \tilde{V}} \right]\Delta u_i + \frac{1}{2}(e_i + E_i)^T Q_i (e_i + E_i) \tag{5.14}$$

The solution of this unconstrained optimization problem must satisfy the following stationary condition.

$$\frac{dJ_i}{d\Delta u_i} = \left(A_{ii}^T Q_i A_{ii} + \underbrace{\tilde{V}_i^T \alpha_i \tilde{V}_i} + R_i \right)\Delta u_i + A_{ii}^T Q_i (E_i) + \underbrace{\tilde{V}_i^T \alpha_i (\eta_i)} + A_{ii}^T Q_i (e_i) = 0 \tag{5.15}$$

Repeating for all the controllers $i = 1, \cdots, N$ produces the following set of linear equations.

$$\begin{bmatrix} A_{11}^T Q_1 A_{11} + \left(\tilde{V}_1^T \alpha_1 \tilde{V}_1\right) + R_1 & A_{11}^T Q_1 A_{12} & \cdots \cdots & A_{11}^T Q_1 A_{1N} \\ A_{22}^T Q_2 A_{21} & A_{22}^T Q_2 A_{22} + \left(\tilde{V}_2^T \alpha_2 \tilde{V}_2\right) + R_2 & \vdots \ \vdots & A_{22}^T Q_2 A_{2N} \\ \vdots & \cdots & \ddots \ \vdots & \vdots \\ \vdots & \cdots & \cdots \ \ddots & \vdots \\ A_{NN}^T Q_N A_{N1} & A_{NN}^T Q_N A_{N2} & \cdots \cdots & A_{NN}^T Q_N A_{NN} + \left(\tilde{V}_N^T \alpha_N \tilde{V}_N\right) + R_N \end{bmatrix} \cdot \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \vdots \\ \Delta u_N \end{bmatrix}$$

$$+ \begin{bmatrix} A_{11}^T Q_1 \cdot e_1 \\ A_{22}^T Q_2 \cdot e_2 \\ \vdots \\ \vdots \\ A_{NN}^T Q_N \cdot e_N \end{bmatrix} + \begin{bmatrix} \tilde{V}_1^T \alpha_1 \cdot \eta_1 \\ \tilde{V}_2^T \alpha_2 \cdot \eta_2 \\ \vdots \\ \vdots \\ \tilde{V}_N^T \alpha_N \cdot \eta_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \tag{5.16}$$

Then, defining the following matrices.

$$\dddot{V} = \begin{bmatrix} \tilde{V}_1^T \alpha_1 \tilde{V}_1 & 0 & 0 & 0 \\ 0 & \tilde{V}_2^T \alpha_2 \tilde{V}_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \tilde{V}_N^T \alpha_N \tilde{V}_N \end{bmatrix} \quad \dot{V} = \begin{bmatrix} \tilde{V}_1^T \alpha_1 & 0 & 0 & 0 \\ 0 & \tilde{V}_2^T \alpha_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \tilde{V}_N^T \alpha_N \end{bmatrix}$$

Equation (5.16) can also be expressed in the following condensed form.

$$\left(A_D^T Q\, A_C + \dddot{V} + R\right)\Delta u + A_D^T Q \cdot (e) + \dot{V} \cdot (\eta) = 0 \tag{5.17}$$

or

$$\Delta u = \left(A_D{}^T Q A_C + R + \ddot{V}\right)^{-1}\left(A_D{}^T Q(-e) + \dot{V}(-\eta)\right).$$

A test case considering the additional control objective is now presented. This time the bottoms reflux temperature, y_3 is not considered a priority, instead maintaining the bottoms reflux duty, u_3 close to -0.05 becomes a control objective. The tuning parameters that reflect these conditions are presented in Table 5.13.

Table 5.13 Modified tuning parameters (Input Reference Control)

Parameter	Description	Value
α	**Input variable weighting**	$[0 \quad 0 \quad 0.1]^T$
Q	Output variable weighting	$[1 \quad 1 \quad 0.0001]^T$
R	Input suppression factor	$[10 \quad 10 \quad 10]^T$

Figure 5.17 illustrates the simulation for this case, basically u_3 is slowly driven to the reference point and the rest of the control objectives are quickly met due to the different weighting factors assigned in the tuning parameters.

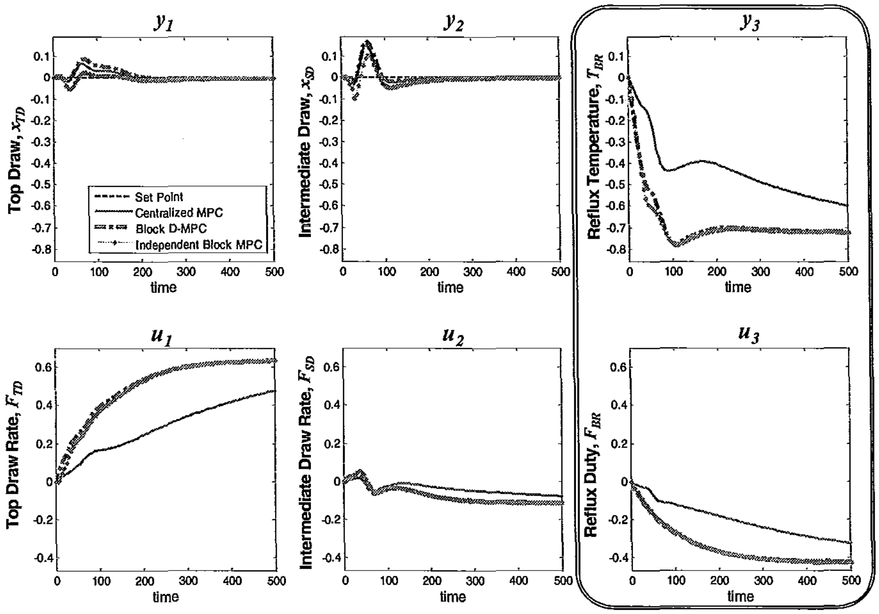


Figure 5.17 Dynamic Performance (Input Reference Control)

5.5 Case Study III: Fired Heater Box.

The final case study considers a fired heater with four valves and four temperatures, and it was first published by Rosenbrock (1974). According to Kariwala et al. (2003) the approximated number of the SISO structure alternatives for this control system is $N_S(4) \approx 4!^{1.52} = 125$. However, in this study a total of only four block structures are considered along with a fully centralized structure. Figure 5.18 shows a diagram of the fired heater and the set of block structures considered for this case.

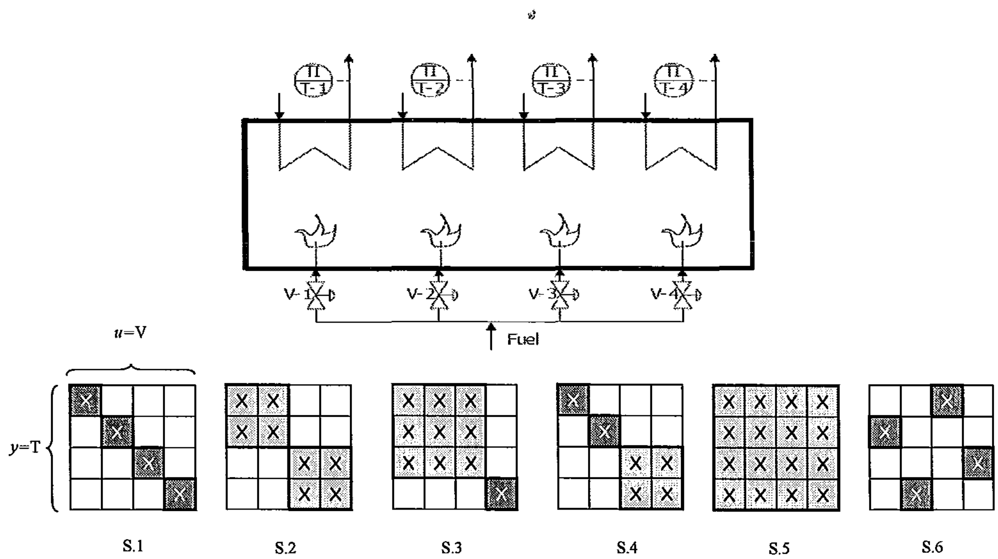


Figure 5.18 Fired Heater Box and Block Structures (Cai, 2009)

Rosenbrock (1974) reported that the dynamic models were obtained by injecting steps signals. However, only half the step responses were measured, and it was assumed that the geometric symmetry of the system would be reflected in the corresponding symmetry properties of the dynamic model. The process and a disturbance model are presented in (5.18). In this case study we treat all variables as dimensionless deviation variables to be consistent with the original citation.

$$G(s)=\begin{pmatrix} \frac{1}{4s+1} & \frac{0.7}{5s+1} & \frac{0.3}{5s+1} & \frac{0.2}{5s+1} \\ \frac{0.6}{5s+1} & \frac{1}{4s+1} & \frac{0.4}{5s+1} & \frac{0.35}{5s+1} \\ \frac{0.35}{5s+1} & \frac{0.4}{5s+1} & \frac{1}{4s+1} & \frac{1}{5s+1} \\ \frac{0.2}{5s+1} & \frac{0.3}{5s+1} & \frac{0.7}{5s+1} & \frac{1}{4s+1} \end{pmatrix} \qquad G_d(s)=\begin{pmatrix} \frac{1}{4s+1} \\ \frac{1}{4s+1} \\ \frac{1}{4s+1} \\ \frac{1}{4s+1} \end{pmatrix} \tag{5.18}$$

The RGA is the following and the block relative gains of the selected structures are shown in Table 5.14.

$$\Lambda = \begin{pmatrix} 1.748 & -0.686 & -0.096 & 0.034 \\ -0.727 & 1.874 & -0.092 & -0.055 \\ -0.055 & -0.092 & 1.874 & -0.727 \\ 0.034 & -0.096 & -0.686 & 1.748 \end{pmatrix}. \quad (5.19)$$

Table 5.14 Block Relative Gain

	Control Structure	BRG
(S.2)	$[y_1 y_2 - u_1 u_2],[y_3 y_4 - u_3 u_4]$	$\lvert\Lambda_{B11}\rvert = 1.213$
(S.3)	$[y_1 y_2 y_3 - u_1 u_2 u_3],[y_4 - u_4]$	$\lvert\Lambda_{B11}\rvert = 1.748$
(S.4)	$[y_1 - u_1],[y_2 - u_2],[y_3 y_4 - u_3 u_4]$	$\lvert\Lambda_{B33}\rvert = 1.213$

The tuning parameters used for the dynamic simulations are described in Table 5.15 and are taken from the thesis by Cai (2009).

Table 5.15 D-MPC tuning parameters

Parameter	Description	Value
m	Input horizon	10
p	Output horizon	60
Δt	Sampling time	0.5
R	Move Suppression factor	$[0.733 \quad 1.13 \quad 1.274 \quad 1.7]^T$
Q	Output variable weighting	$[1 \quad 1 \quad 1 \quad 1]^T$

The dynamic simulation is subject to a step set-point change in temperatures T_2, T_3 and T_4 while T_1 is subject to a first order dynamics set point change. Figure 5.19 shows the closed loop simulation. The objective is to identify which block structure with a D-MPC controller provides the best dynamic performance.

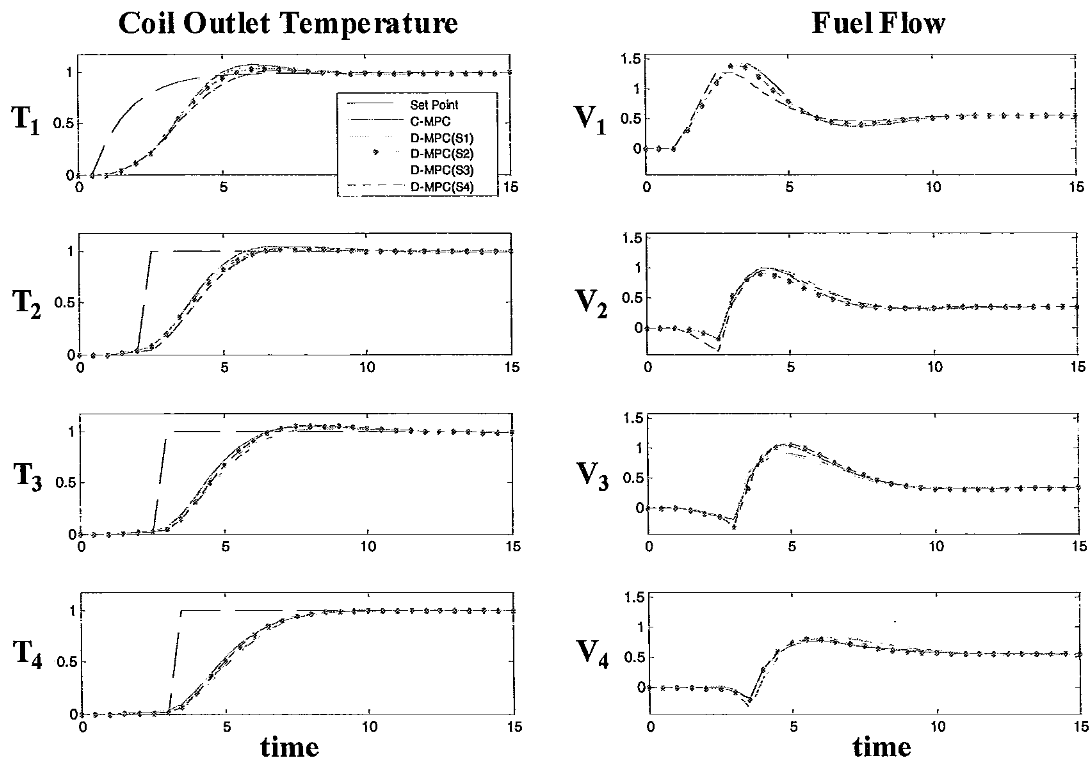


Figure 5.19 Dynamic Performance – Set Point Change with Positive BRG

The results in Table 5.16 basically indicate that the D-MPC with a diagonal structure (S.1) provides the worst performance while the D-MPC with structure (S.3) gets closer to the centralized MPC.

Table 5.16 Integral of the Squared Error, Set Point Changes

Output Variable	C-MPC **S.5**	D-MPC **S.1** [T1-V1][T2-V2] [T3-V3][T4-V4]	**S.2** [T1T2-V1V2] [T3T4-V3V4]	**S.3** [T1T2T3 -V1V2V3] [T4-V4]	**S.4** [T1-V1][T2-V2] [T3T4-V3V4]
$y_1(T_1)$	0.753	0.797	0.752	0.749	0.795
$y_2(T_2)$	1.094	1.306	1.128	1.103	1.294
$y_3(T_3)$	1.130	1.335	1.244	1.170	1.287
$y_3(T_4)$	1.099	1.334	1.182	1.250	1.201
Total	**4.077**	**4.773**	**4.306**	**4.272**	**4.576**

A second case is also considered, this time the same set of structures are subject to a disturbance change $[l_1 = 0.5]$. Figure 5.20 illustrates the closed loop simulation while Table 5.17 shows the performance index.

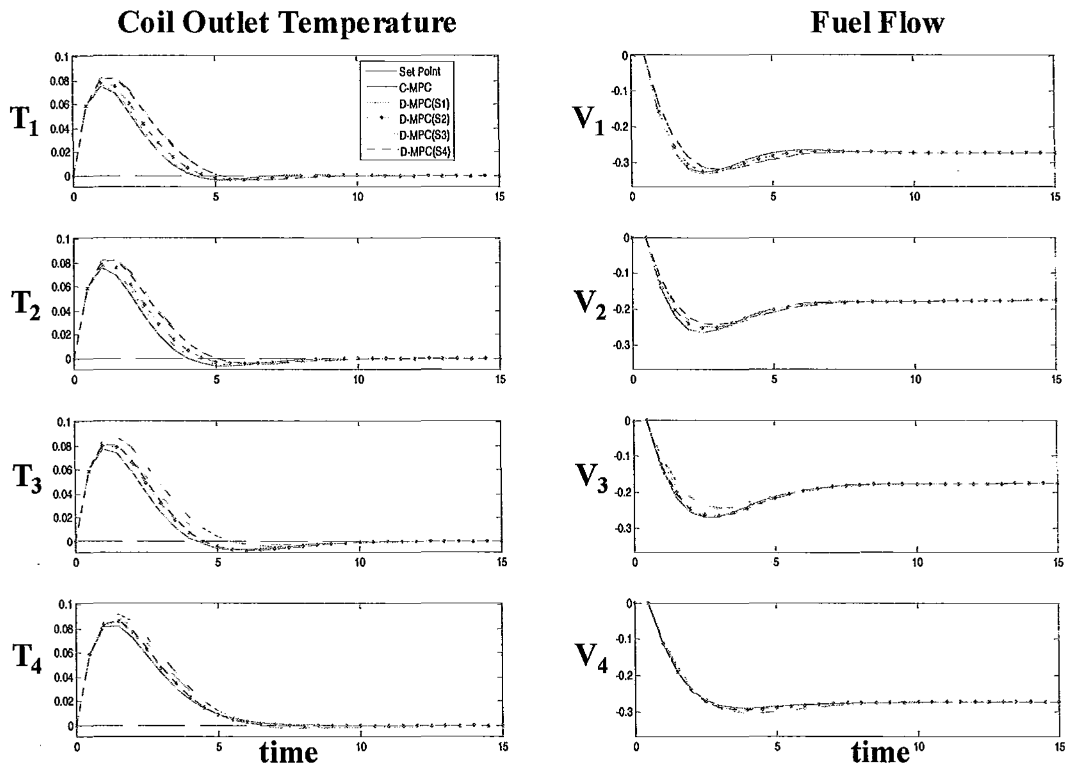


Figure 5.20 Dynamic Performance - Disturbance Change with Positive BRG

This time the block structure with the closest ISE index to the C-MPC controller is structure S.2 that consists of a couple of *two by two* controllers.

Table 5.17 Integral of the Squared Error, Disturbance Change [l_1=*0.5*]

Output Variable	C-MPC S.5	D-MPC S.1 [T1-V1][T2-V2][T3-V3][T4-V4]	D-MPC S.2 [T1T2-V1V2][T3T4-V3V4]	D-MPC S.3 [T1T2T3 -V1V2V3][T4-V4]	D-MPC S.4 [T1-V1][T2-V2][T3T4-V3V4]
$y_1(T_1)$	0.0092	0.014	0.011	0.0098	0.0137
$y_2(T_2)$	0.0093	0.014	0.0112	0.0103	0.0134
$y_3(T_3)$	0.0106	0.0162	0.0122	0.0133	0.0128
$y_3(T_4)$	0.0145	0.0194	0.0159	0.0178	0.0163
Total	**0.0436**	**0.0637**	**0.0503**	**0.0512**	**0.0562**

Finally, a negative BRG structure (S.6) is considered. In this case the tuning reported previously produces an unstable controller; therefore, some changes are required in order to stabilize the process. In order to minimize the amount of parameters to be modified the suppression factor is adjusted by applying a multiplication factor, $\beta = 3.4$. A stability analysis of this design was performed confirming nominal stability. Figure 5.21 shows the closed loop simulation.

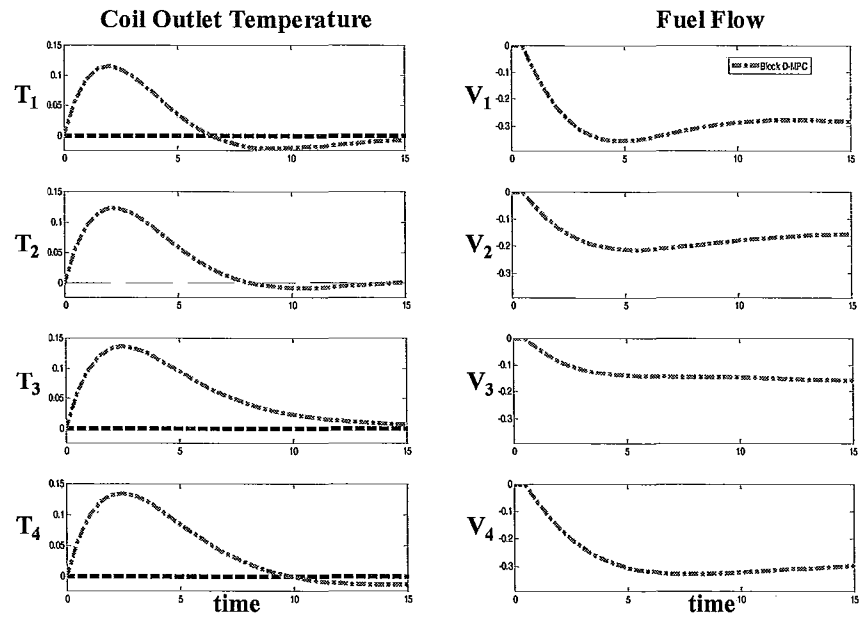


Figure 5.21 Dynamic Performance - Disturbance Change with Negative BRG

Although the process is stabilized the negative BRG structure produces the worst dynamic performance with a total ISE=0.647.

From the cases above it can be observed that structure S.2 is better for disturbance rejection while structure S.3 performs better when subject to a specific group of set point changes.

5.6 Summary

In this chapter the proposed D-MPC controller was tested in three different case studies. All of the simulation performed for each case study used the same set of tuning parameters in order to have a base for comparison; however, and because each the C-MPC and the D-MPC controller have different goals it is important to mention that a suitable measure for comparison will require a robust control analysis which is beyond the scope of this research.

An important advantage presented is that the unconstrained D-MPC could easily test the implementation of different control structure by performing a simply modification of matrix A_D, which pretty much takes the form of the block structures as described in Figure 5.9 and Figure 5.18. Then, by testing different control structures for the D-MPC it was confirmed that the closer the control structure gets to the centralized structure, the better the performance measured by ISE. However, the closer it gets to the centralized structure the farthest it deviates from local autonomy. In this way and in order to design a suitable control system a trade off must be evaluated.

The D-MPC controller outperformed independent block MPC and in cases with negative and zero BRG it surpassed the capabilities of current iterative methods for coordinated block MPC. However, in the case of constrained D-MPC it also showed a situation where the KKT approach described in Section 3.3.2 may encounter difficulties and therefore fail to provide a satisfactory response. Although some adjustments of the solver with additional strategies such as warm start might help in some circumstances, in this work it was preferred to implement a heuristic D-MPC strategy that could easily overcome such situations and provide tractable and reliable real-time computation.

Finally, the addition of alternative control objectives was also addressed and the stability analysis developed in Chapter 4 was validated with closed loop simulations for different control structures.

Chapter 6

Conclusions and Future Work

This chapter presents a brief summary of the work presented, followed by the main contributions. The final section introduces some possible future research directions.

6.1 Summary

In this thesis, a novel block decentralized MPC approach is developed in order to coordinate the control of individual process units (blocks) in a chemical plant. Chapter 1 addressed the problem definition, control objectives and the anticipated significance of the research. It clearly pointed out that even in these times of almost "unlimited" computational resources, decentralized MPC control is needed for block autonomy, disturbance isolation, fault tolerance, ease of management and most important, dynamic performance.

In Chapter 2 the state of the art methods for the coordination of block MPC controllers were reviewed. The review distinguished two major categories: (1) the methods that implement a block design and enforce autonomy and (2) the methods that implement a centralized design and whose main objective is to emulate a centralized MPC controller by using some sort of distributed computational architecture. It also introduces the reader to the key concepts required to understand the goals of the research such as: integrity, local autonomy and dynamic performance.

In Chapters 3 and 4 the conceptual developments of this research were presented. In Chapter 3 the D-MPC controller is formulated by building on the QDMC algorithm (Garcia and Morshedi, 1986). The approach replaces multiple optimizations (from several, interacting MPC controllers) with one set of equations, yielding a single-level optimization problem. The implementation of the D-MPC controller makes use of the concepts of game theory, multilevel optimization, interior point methods and active set strategies. Some simple guidelines were developed to identify when D-MPC is appropriate; however, it was not possible to produce a set of explicit rules that guarantee the existence and uniqueness of the controller. Therefore in Chapter 4 some requirements for the implementation of D-MPC were introduced such as benchmark (C-MPC) applicability, minimum prediction horizon and steady state controllability. Then, in the last part of Chapter 4 a nominal stability analysis was successfully adapted for the D-MPC controller. All of this analysis was integrated into a step-wise application procedure to generate a non-singular and stable D-MPC controller with satisfactory dynamic performance.

Finally in Chapter 5 all of the previous concepts and methodologies were evaluated in several cases studies, each considering multiple block control structures. The case studies confirmed the premise that the D-MPC produces a better performance than the independent block MPC. Additionally it also showed the ease with which multiple control structures can be implemented and the superiority of the proposed controller over other communication MPC methods.

6.2 Contributions

This section presents the specific contributions of this research.

- **Formulation of the D-MPC Controller:** The D-MPC controller makes use of local objective functions that are optimized in each block controller by adjusting only its manipulated variables. Additionally, interaction models are embedded in each controller. This provides the local autonomy sought while accounting for interactions. Due to the interaction models, processes with dominant interactions $(|BRG| \leq 0)$ can be successfully controlled using D-MPC. This distinctive characteristic extends the range of processes and block designs for which block MPC is possible.

- **Solution Algorithm:** The D-MPC controller applies a strategy similar to that of multilevel optimization. Here, several optimization problems (Block MPC) are replaced with their respective optimality conditions and solved simultaneously. An advantage of this method is that overcomes the necessity of an iterative algorithm. A disadvantage is the non-convexity of the problem, which makes assurance of a correct solution problematic for formulations including inequality constraints, especially with negative and zero BRG configurations. A modified formulation was developed to deal with inequality constraints, which removes the non-convexity by implementing an active-set heuristic strategy to enhance the D-MPC formulation. The approach is computationally tractable yielding a small set of convex problems to be solved sequentially; however, global optimality cannot be assured.

- **Design and Analysis:** One analysis demonstrated that autonomous (D-MPC) performance cannot be achieved through tuning a conventional centralized (C-MPC) controller. The next step determined the applicability of the D-MPC controller. It was concluded that minimum criteria are required for the D-MPC application, therefore it was proposed that: (1) only stable processes that can be controlled by C-MPC would be considered for D-MPC control, (2) since applications are for continuous processes that track their set points, the process must be steady-state controllable, with the column rank of the gain matrix being equal to the number of controlled variables, (3) finally, it was found that prediction horizon of the controller influences the controllability, and a minimum output horizon was suggested.

Once the existence of the controller has been achieved the next step is to guarantee the stability. The analysis of nominal stability of the D-MPC controller is another important contribution of this work. The stability analysis is adapted from the work of Lee et al (1994) to analyze the closed loop stability of different block structures. It is also applicable to the other controllers considered in this work (i.e., independent block MPC, extended D-MPC and of course C-MPC).

Finally, the design of the D-MPC controller is addressed by means of a step-wise application procedure that ultimately ensures a successful application: First, the steady-state controllability and the existence of a C-MPC are verified. Then, tuning adjustment is required in order to produce a nonsingular and nominally stable D-MPC controller. Finally, simulations are performed in order to fine-tune the controller and provide the desired dynamic performance.

- **Case Studies:** The capabilities of the proposed D-MPC controller were tested for 2x2 to 4x4 cases where it was demonstrated that:

- Effects (i.e., offset) of manipulated variable saturation are isolated to the blocks where saturation occurs.

- D-MPC successfully control block structures with positive, negative and zero BRG.

- Nominal Stability can depend on tuning.

- Good nominal dynamic performance is achieved for all cases.

- For Single Horizon D-MPC controller, processes paired on positive RGA, a unique stable solution for any controllable plant, $(|Kp| \neq 0)$ always exists; and processes paired on negative RGA present both a stable (negative feedback) and an unstable (positive feedback) zone.

6.3 Future Work

The D-MPC proved to provide good performance for disturbances and set point changes while enforcing local autonomy. Moreover different control structures with different integrity and even different objective functions are easily coordinated. In order to further enhance the D-MPC controller there are several research opportunities to pursue. The author considers the following the most important future directions.

- **Inclusion of Soft output constraints**: The very next step to enhance the D-MPC controller is the inclusion of soft output constraints to the heuristic strategy. Taking into account soft output constraints results in a challenging problem. These constraints include a positive slack variable that expands the bound if necessary $(y \leq y_{\max} + s)$. This slack value has to be included in the algorithm and its value cannot be easily fixed a priori. None of the other heuristic methods used in industry (Prett et al 1980, and Richalet et al 1987) proposed a systematic method to solve this problem. In order to solve this problem alternative methods must be considered.

- **Model Uncertainty**: An important issue that needs to be addressed is the uncertainty in the process model. In order to tackle this problem concepts of Robust MPC are necessary. Properties such as robust stability and performance could be adapted for the decentralized environment. Al-Gherwi et al (2008) proposed a methodology to select the best block structure based on robust performance. The resulting structure will be more resilient to model mismatch.

- **Coordination of Steady-State MPC:** While autonomy is provided to each controller when solving the dynamic problem, it may or may not be desired in the steady-state coordination problem. If autonomy is desired at the steady-state level, the previously described methodology could be extended to the steady state level. In fact if the steady state optimizers are formulated as QP problems, the results from the Single Horizon D-MPC might provide some insight and a head start regarding singularity and stability. However if the overall plant is to be optimized, an approach similar to current technology with a global steady state coordinator could be applied. The choice would be application dependent.

- **State Space Formulation**: Finally, another important improvement will be the development of a state space formulation to provide control for open loop unstable processes. Such formulation will also allow enhancing the simple state estimation approach by adding a Kalman filter. It is thought that the migration of D-MPC to a state space formulation could be relatively straightforward.

Nomenclature

$A_{i,j}$	Dynamic matrix with the effect on outputs in block i from inputs in block j
$\hat{A}_{gf}$	Dynamic matrix that relates output g and input f
A_C	Full dynamic matrix
A_D	Block diagonal dynamic matrix
a	Step response coefficient
A_S	Step response model
A^d	Step response coefficients for disturbance model
B_i	Vector of maximum changes in block i
d_i	Vector of unmeasured disturbance in block i
E_i	Effect of interactions in block i
e_i	Vector of feedback information in block i
H_i	Matrix of active constraints in block i
H_{SS}	Closed loop matrix for stability analysis
I	Identity matrix of appropriate size
J_i	Objective function in block i
K_{MPC}	Control gain matrix for centralized MPC
K_{DMPC}	Control gain matrix for Block D-MPC
Kp_{gf}	Steady state gain that relates output g and input f
L_i	Lagrange function in block i
m_i	Input horizon in block i
M_i	Number of manipulated variables in block i
$\dot{M}_{jj}$	Minor of element Kp_{jj}
N	Number of blocks to be coordinated
p_i	Prediction horizon in block i

P_i	Number of controlled variables in block i
Q_i	Weight factor for outputs in block i
R_i	Suppression factor for inputs in block i
$SS_{\max}$	Required samples to reach steady state of the slowest process.
$u_i^{\max}, u_i^{\min}$	Bounds on input variables in block i
u_i^{ref}	Vector of input reference in block i
u_i	Vector of input variables in block i
V	Lower triangular matrix of ones.
$\widetilde{V}$	Block diagonal matrix built from V matrices
$W_{i,j}$	Weighting factors for outputs in block $j \neq i$ to be used in the objective function of block i
$X(k)$	Dynamic state vector at time k
$X_{SP}(k)$	Output reference vector
y_i *	Vector of (measured) controlled variables in block i
y_i^{sp}	Vector of output set points in block i
y_i^p	Vector of past output in block i
y_i	Vector of predicted values in block i

Subscripts

i, j	Refers to control blocks.
f	Refers to input variables.
g	Refers to output variables.
k	Refers to a time step
ℓ	Refers to a time step in prediction horizon
ff	Refers to an element of the input variable vector

Greek letter

λ_i, ϕ_i	Lagrange multipliers in block i
α_i	Weighting factor for input variables in block i

v	Vector of eigenvalues
ε	Small gain mismatch
$\Lambda_{g,f}$	Relative gain that relates output g and input f
$\Lambda_{B\,i,j}$	Block Relative Gain that relates block i and block j
Δu_i	Vector of input moves in block i
$\Delta u_i^{\max}, \Delta u_i^{\min}$	Bounds on input size variables in block i

Acronyms

BRG	Block Relative Gain
CV	Controlled Variables
KKT	Karush Kuhn Tucker
LP	Linear Programming
LTI	Linear Time Invariant
MIMO	Multiple Input Multiple Output
MPC	Model Predictive Controller
MV	Manipulated Variables
PD	Positive Definite
PID	Proportional Integral Derivative
PSD	Positive Semidefinite
QDMC	Quadratic Dynamic Matrix Control
QP	Quadratic Programming
RGA	Relative Gain Array
SISO	Single Input Single Output

References

Al-Gherwi, W., Budman, H. and Elkamel, A. (2008). Robustness Issues Related to the Application of Distributed Model Predictive Control Strategies. The International Federation of Automatic control. Seoul, Korea, July 6-11, 2008.

Baker, R. (2006). Interior Point Methods in Integrated Design and Control. PhD. Thesis, McMaster University. Hamilton On Canada.

Baker, R. and Swartz, C.L.E (2008) Interior Point Solution of Multilevel Quadratic Programming Problems in Constrained Model Predictive Control Applications. Ind. Eng. Chem. Res., 2008, 47 (1).

Bartlett, R. and Biegler, L.T. (2006). QPSchur: A Dual, Active-Set, Schur Complement Method for Large-Scale and Structured Convex Quadratic Programming. Optimization Engineering. 7, 5-32.

Başar, T. and Olsder, G. J. (1982). Dynamic Noncooperative Game Theory. Academic Press Inc. Ltd. London.

Biegler, L. T., Grossman, I.E. and Westerberg, A. W. (1997). Systematic Methods of Chemical Process Design. Prentice Hall.

Bristol, E.H (1966). On a New Measure of Interaction for Multivariable Process Control. IEEE Trans. Automatic Control. 11, 133.

Brosilow, C. and Joseph, B. (2002). Techniques of Model-Based Control. Prentice Hall.

Cai, Y. (2009) Systematic Approach for Control Structure Design. PhD. Thesis, McMaster University. Hamilton On Canada.

Campo, P.J., and Morari, M. (1994). Achievable Closed-Loop Properties of System Under Decentralized Control: Conditions Involving the Steady-State Gain. IEEE Transactions on Automatic Control. 39, No. 5, 932-943.

Camponogara, Eduardo, Dong Jia, Bruce H. Krogh and Sarosh Talukdar (2002). Distributed Model Predictive Control. IEEE Control Systems Magazine. February 44-52.

Charos, G.N., and Arkun, Y. (1993). A Decentralized Quadratic Dynamic Matrix Control Algorithm, Journal of Process Control, 3, 75-83.

Cheng R., Forbes J.F. and Yip, W.S. (2007). Price-driven Coordination Method for Solving Plant-wide MPC Problems. Journal of Process Control (17) pp. 429-438

Cheng, R. and Forbes, J.F. (2004). Dantzing-Wolfe Decomposition and Large-Scale Constrained MPC Problems. In Proceedings of the 7th IFAC Symposium on Dynamic and Control of Process Systems (DYCOPS-7).

Chiu, M.S. and Arkun, Y. (1990). Decentralized Control Structure Selection Based on Integrity Considerations. Industrial & Engineering Chemistry Research, 29, No. 3. 374-382.

Clark, P.A. and Westerberg, A.W. (1990). Bilevel Programming for Steady State Chemical Process Design I. Computers and Chemical Engineering, 14(1), 87-97.

Colson, B., Marcotte, P. and Savard, G. (2005). Bilevel Programming: A Survey. 4OR: A Quarterly Journal of Operations Research, 3, No. pp 87-107.

Cutler, C.R. and Ramaker, B.L. (1980). Dynamic Matrix Control: A Computer Control Algorithm. Proceedings Joint Auto Control Conference, Paper Wp5-B, San Francisco.

Du, X., Xi, Y. and Li, S.. (2001). Proceedings of the American Control Conference. Arlington, VA. 3142-3143.

Garcia, C. E., and Morari, M. (1982). Internal Model Control, 1: A unifying Review and some New Results. IEC Proc. Des. Dev., 21, 73.

Garcia, C.E. and Morshedi, A.M. (1986). Quadratic Programming Solution to Dynamic Matrix control (QDMC). Chemical Engineering communication. 46, 73.

Giovanini, L., Balderud, J. and Katebi, R. (2007). Autonomous and Decentralized Mission Planning for Clusters of UUVs. International Journal of Control. Vol 80, No. 7, 1169-1179

Hoffman, D.W. (2000). Process Control: Achieving the Benefit. International Symposium on Advanced Control of Chemical Processes. Pisa Italy, June 14-16, 2000.

Horn, R.A. and Johnson, C.R. (1976). Matrix Analysis. Cambridge University Press.

Jakhete, R. Rager, W. and Hoffman, S.W. (1999). Online Implementation of Composite LP Optimizes FCCU/GPU Complex, Hydrocarbon Processing, February 1999.

Jia. D. and Krogh, B.H. (2001). Distributed Model Predictive Control. Proceedings of the American Control Conference. Arlington, VA. 2767-2772.

Kakalis, N., Dua, V., Sakizlis, V., Perkins, J. D., and Pistikopoulos, E. N. (2004). Design of Robust Model-based Controllers Via Parametric Programming. Automatica 40 189 – 201

Kariwala, V., Forbes, J.F. and Meadows E. S..(2003) "Block Relative Gain: Properties and Pairing Rules." Industrial and Engineering Chemistry Research 42, no. 20. 4564-4574.

Lee, J.H., Morari, M. and Garcia, C. E. (1994). State-Space Interpretation of Model Predictive Control. Automatica 30 No. 4 pp 707-717.

Li, S., Lim. K.Y., and Fisher, D.G. (1989). A State Space Formulation for Model Predictive Control. AICHE Journal. 35, No 2. pp 241-249.

Li, S., Zhang, Y. and Zhu, Q. (2005). Nash-optimization Enhanced Distributed Model Predictive Control Applied to the Shell Benchmark Problem. Information Sciences (170) pp 329-349.

Lu, J. Z. (2001). Challenging Control Problems and Emerging Technologies in Enterprise Optimization. In Proceedings of the 6th IFAC Symposium on Dynamic and Control of Process Systems (DYCOPS-6) (23–34).

Luyben, W.L., Tyreus, B.D. and Luyben, M.L. (1998). Plantwide Process Control. McGraw Hill.

Manousiouthakis, V., Savage, R., and Arkun, Y. (1986). Synthesis of Decentralized Process Control Structures Using the Concept of Block Relative Gain. AIChE Journal, 32(6), 991–1003.

Marlin, T.E. (2000). Process Control: Designing Processes and Control Systems for Dynamic Performance. McGraw-Hill (2nd Edition).

Nath, R. and Alzein, Z. (2002). On-line Dynamic Optimization of Olefins Plants. Computers and Chemical Engineering, 24, pp 533-538.

Negenborn R.R., De Schutter, B. and Hellendorm, J., (2004) Multi-Agent Model Predictive Control: A Survey. Technical Report. Delft Centre for Systems and Control. Delft, The Netherlands.

Ogunnaike, B. A., and Ray, W. H. (1994). Process Dynamics, Modeling, and Control. Oxford University Press, New York.

Osbourne, M.J, Rubinstein, A. (1994). A Course in Game Theory. MIT Press.

Prett, D.M. and Gillete, R.D. (1980). Dynamic Optimization and Constrained Multivariable Control of a Catalytic Cracking Unit. 1980 Joint Automatic Control Conference

Prett, D.M. and Morari, M., (1987) The Shell Process Control Workshop, Butterworths, Boston.

Qin, S. J., and Badgwell, T. A. (2003). A Survey of Industrial Model Predictive Control Technology. Control Engineering Practice, 11, 7, 733-764

Raghunathan, A. U. and Biegler, L.T. (2003). Mathematical Programs with Equilibrium Constraints (MPECs) in Process Engineering. Computers and Chemical Engineering, 27, 1381-1392.

Richalet, J.A., Rault, J.L., Testud and Papon J. (1987). Model Predictive Heuristic Control: Application to Industrial Processes. Automatica (14). Pp 413-428.

Rosenbrock (1974) Computer-aided Control System Design. New York: Academic Press.

Seborg, D. E., Edgar, T.F. and Mellichamp, D.A. (2003). Process Dynamics and Control. John and Wiley & Sons. (2nd Edition)

Shigueo-Hori, E. and Hong W. (2005) Coordinated Control of Decentralized QDMC Controllers Applied to a Reactor/Distillation Column with Recycle. In Proceedings of the 2nd Mercosur congress of Chemical Engineering.

Shridhar, R. and Cooper, D. J. (1998). A Tuning Strategy for Unconstrained Multivariable Model Predictive Control Ind. Eng. Chem. Res. 1998. 37, 4003-4016.

Skogestad and Postlethwaite, (1996). Multivariable Feedback Control. John Wiley & Son Ltd.

Stephanopoulos, G. (1983). Synthesis of Control Systems for Chemical Plants – A Challenge for Creativity. Computers and Chemical Engineering, 7, 331-365.

Strang G. (1980). Linear Algebra and its Application. Academic Press.

Van Schuppen, J.H. (2004). Decentralized Control with Communication Between Controllers, Unsolved Problems in Mathematical Systems and Control Theory, Vincent D. Blondel, Alexander Megretski (Eds.), Princeton University Press, Princeton, 2004, pp. 144–150.

Venkat, A.N., Rawlings, J.B. and Wright S.J. (2004). Plant-wide Optimal Control with Decentralized MPC. In Proceedings of the 7th IFAC Symposium on Dynamic and Control of Process Systems (DYCOPS-7).

Wagler, Randy (2007). Profit Suite Layered Optimization. MACC Workshop May 2007, McMaster University, Hamilton On Canada.

Warren, Adam (2004). Sequential Decision-Making Under Uncertainty. PhD Thesis McMaster University, Hamilton On. Canada.

Zhang, Y. and Li, S. (2007). Networked Model Predictive Control Based on Neighbourhood Optimization for Serially Connected Large-scale Processes. Journal of Process Control, 17, 37–50

Appendix A. Model Predictive Control

This appendix addresses the basic Model Predictive Control algorithm used in this thesis. The main algorithm along with the most important assumptions and the tuning parameters are described. It is important to mention that the entire treatment of this controller is adopted in the development of the D-MPC controller. This presentation follows the explanation in Brosilow and Joseph (2002).

A.1 Mathematical Models in MPC

The way in which dynamic models are handled in any MPC methodology is one of the most important aspects of the control algorithm. In this section the dynamic matrices used in the proposed controller are presented to demonstrate the mathematical framework behind the MPC models.

The linear MPC algorithm developed in this work is based on step-weight models. A Single Input Single Output (SISO) step-weight model has the following form.

$$\begin{bmatrix} y_{g,(k+1)} \\ \vdots \\ y_{g,(k+p)} \end{bmatrix} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \ddots & 0 \\ \vdots & . & . & 0 \\ a_p & a_{p-1} & \cdots & a_{p-m} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_{f,(k)} \\ \vdots \\ \Delta u_{f,(k+m-1)} \end{bmatrix} \tag{A.1}$$

Here a vector, $\Delta u_f \in \Re^m$ representing the m future input moves of **input variable** f is related to the future p output changes, $y_g \in \Re^p$ of **output variable** g through the step-weight matrix $\hat{A}_{gf} \in \Re^{pxm}$. The expression can be expressed as follows.

$$y_g = \hat{A}_{g,f} \cdot \Delta u_f \quad (A.2)$$

Where each input move is defined as: $\Delta u_{f,(k)} = u_{f,(k+1)} - u_{f,(k)}$ and y_g is given as a deviation variable that represents the deviation from the current steady-state.

This model gets its name from the fact that the first column of matrix $\hat{A}_{g,f}$ represents the response of the system to a unit step change in u at time, $t = 0$.

Figure A.1 shows the response of a first-order system to a unit step change in u. This response is compared with the values of the first column of the step-weight matrix that represents this process i.e. the step-weights. The solid line shows the continuous output response and the step weights are shown as bars.

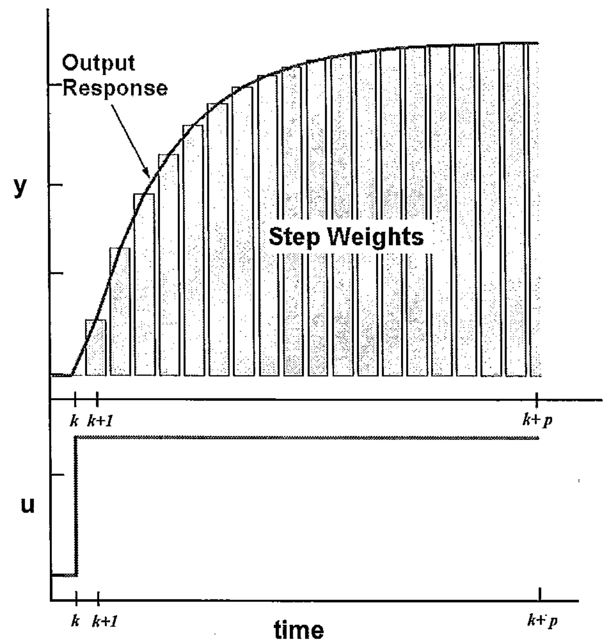


Figure A.1 Comparison of Step-Weights to the Continuous Step Response

Using the step response coefficients we can predict changes in y that are caused by any control moves.

$$\begin{aligned} y_{g,(k+1)} &= a_1 \cdot \Delta u_{f,(k)} \\ y_{g,(k+2)} &= a_2 \cdot \Delta u_{f,(k)} + a_1 \cdot \Delta u_{f,(k+1)} \\ &\vdots \\ y_{g,(k+p)} &= a_p \cdot \Delta u_{f,(k)} + \cdots + a_{p-m} \cdot \Delta u_{f,(k+m-1)} \end{aligned} \tag{A.3}$$

The step-weight matrix, $\hat{A}_{g,f}$ is also called the **Dynamic Matrix** of the process. Observe that the number of rows represents the prediction horizon, p while the number of columns represents the input horizon, m or future input moves. The execution time, Δt and p are usually selected so that y_g achieves steady state at the end of the horizon; thus, in most cases a_p equals the steady-state gain.

A.2.1 MPC Models for Multiple Input Multiple Output (MIMO) Systems

The same concept presented for SISO systems is easily extended to multivariable systems. The basic equations remain the same, except that the matrices and vectors become larger and properly partitioned.

Consider the multivariable process in Figure A.2.

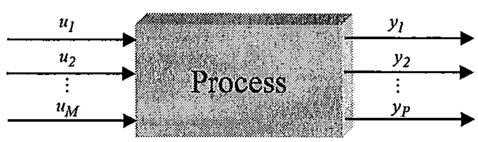


Figure A.2 Multiple Input Multiple Output Process

The relation between outputs y and inputs u for a MIMO case can be modeled as follows.

$$\Delta y = A \cdot \Delta u \tag{A.4}$$

Where $A \in \Re^{pP x mM}$ is now a multivariable dynamic matrix that includes P output variables and M input variables. Each block element is a SISO dynamic matrix that can be expressed as follows.

$$\hat{A}_{gf} = \begin{bmatrix} a_{gf_1} & 0 & \cdots & 0 \\ a_{gf_2} & a_{gf_1} & \ddots & 0 \\ \vdots & \cdot & \cdot & 0 \\ a_{gf_p} & a_{gf_{p-1}} & \cdots & a_{gf_{p-m+1}} \end{bmatrix} \tag{A.5}$$

Where $\hat{A}_{gf} \in \Re^{(p x m)}$ is the dynamic matrix relating the changes in the g^{th} output to the f^{th} input. The MIMO Dynamic Matrix A will now have the following structure.

$$A = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} & \cdots & \hat{A}_{1M} \\ \hat{A}_{21} & \hat{A}_{22} & \cdots & \hat{A}_{2M} \\ \cdots & \cdots & \ddots & \cdots \\ \hat{A}_{P1} & \cdots & \cdots & \hat{A}_{PM} \end{bmatrix} \tag{A.6}$$

Finally the input and output vectors are properly handled in a stacked form.

$$\Delta u = \begin{bmatrix} \Delta u_{1,(k)} \\ \vdots \\ \Delta u_{1,(k+m-1)} \\ \hdashline \Delta u_{2,(k)} \\ \vdots \\ \Delta u_{2,(k+m-1)} \\ \hdashline \vdots \\ \hdashline \Delta u_{M,(k)} \\ \vdots \\ \Delta u_{M,(k+m-1)} \end{bmatrix} \qquad y = \begin{bmatrix} y_{1,(k+1)} \\ \vdots \\ y_{1,(k+p)} \\ \hdashline y_{2,(k+1)} \\ \vdots \\ y_{2,(k+p)} \\ \hdashline \vdots \\ \hdashline y_{P,(k+1)} \\ \vdots \\ y_{P,(k+p)} \end{bmatrix} \tag{A.7}$$

A.2 Overview of the Model Predictive Control

The term Model Predictive Control does not designate a specific control strategy but a range of control methods, which make an explicit use of a model of the process to obtain the control signal by minimizing an objective function. Basically an MPC algorithm is composed of the following (Camacho and Bordons, 1999):

- Explicit use of a dynamic model to predict process output at future time instants.
- Calculation of a control sequence minimizing an objective function.
- Use of a receding strategy, where the only the first input move of the control sequence is implemented and then the controller is resolved at the next execution time.

Figure A.3 illustrates the two main steps of the basic MPC strategy, (1) First at time, k, the controller uses past information to predict the future, "open-loop" behavior of the process output. The output prediction assumes that future control actions are *zero* and includes the predicted effects of past input moves, measured disturbances and a feedback estimate of unmeasured disturbances. (2) Then the controller calculates a set of input moves that will minimize an objective function that in some manner measures the difference between the predicted output and the output set points.

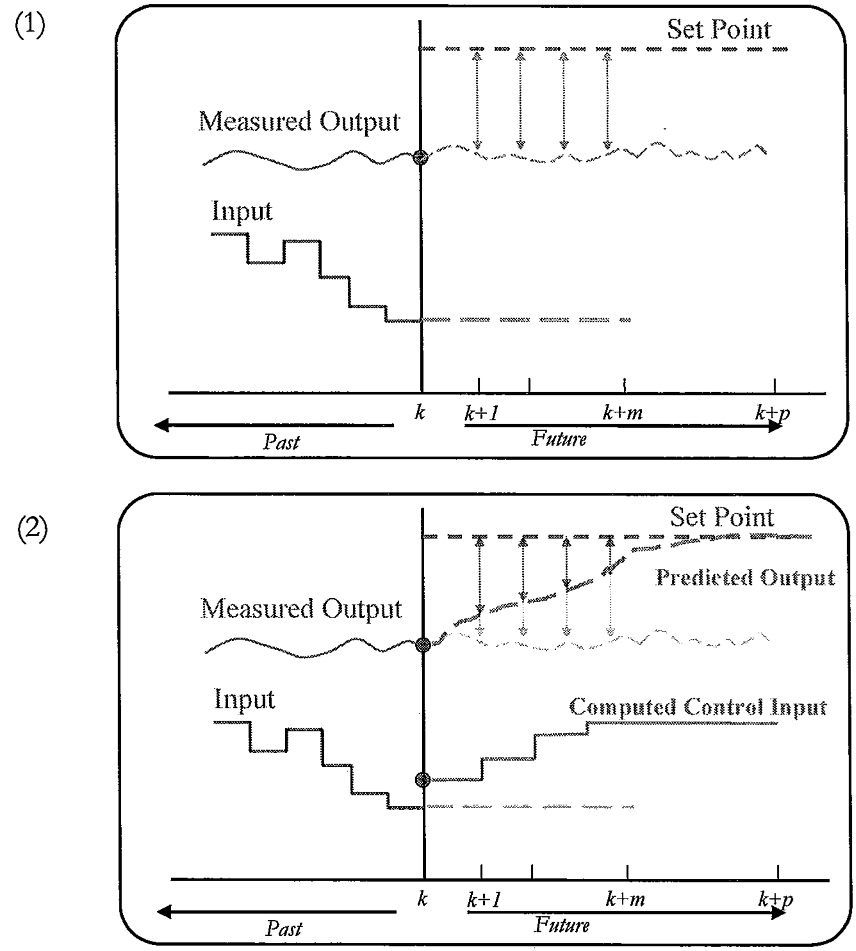


Figure A.3 Basic steps for MPC Strategy

A.3.1 Description of the MPC Algorithm

Let's begin the treatment of the MPC algorithm by illustrating the block diagram representation in Figure A.4.

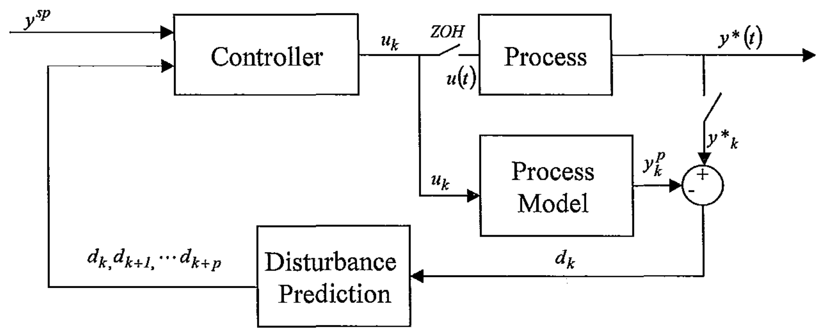


Figure A.4 Block Diagram of the MPC Algorithm

At any arbitrary sample time, denoted by k, the problem can be stated as follows. **Given** a desired output trajectory $y^{sp}_{k+1}, y^{sp}_{k+2}, \cdots, y^{sp}_{k+p}$, **compute** a predicted trajectory of the output $\left(y^p_{k+1}, y^p_{k+2}, \cdots, y^p_{k+p}\right)$ based on past inputs. Then **compute** an estimate of the disturbances $d_{k+1}, d_{k+2}, \cdots, d_{k+p}$ and finally **compute** the control actions $(\Delta u_k, \Delta u_{k+1}, \cdots, \Delta u_{k+m})$ needed to bring the output to the desired trajectory and an estimate of the output trajectory, assuming that the current control action Δu_k is implemented.

Once all the parameters and variables are specified the MPC algorithm begins with the open loop prediction of the future output using past input information. This vector, y^p contains the predicted values of the output over the horizon p and captures the state of the system. This prediction can be calculated as shown below.

$$y^p = \begin{bmatrix} y^p_{k+1} \\ \vdots \\ y^p_{k+p} \end{bmatrix} = \begin{bmatrix} a_2 & a_3 & \cdots & a_{p+1} \\ a_3 & a_4 & \cdots & a_{p+2} \\ \vdots & \vdots & . & \vdots \\ a_p & a_{p+1} & \cdots & a_{p+n_{ss}} \end{bmatrix} \cdot \begin{bmatrix} \Delta u_{k-1} \\ \vdots \\ \Delta u_{k-n_{ss}} \end{bmatrix} \tag{A.8}$$

It is important to note that this calculation requires storing a number of past input values. Also, to limit the stored past measurements to a reachable size, Δu inputs for $k + n_{ss}$ are summed and then multiplied by the steady-state gain. Where n_{ss} refers to the closed-loop settling time of the process.

The next step is to estimate the current disturbance. Here the estimate is evaluated by calculating the difference between the prediction of y_k^p and the current measured value y^*_k.

$$d_k = y^*_k - y_k^p \tag{A.9}$$

One specific manner for estimating future disturbances commonly employs the assumption that the current error between the plant and the model will remain unchanged throughout the output horizon.

$$d_{k+1}, d_{k+2}, \cdots, d_{k+p} = d_k \tag{A.10}$$

Then using the Dynamic Matrix model, $\hat{A} \in \Re^{pxm}$ the future output, $y \in \Re^p$ can be calculated using the past predictions, $y^p \in \Re^p$ the disturbance estimates, $d \in \Re^p$ and the future control actions $\Delta u \in \Re^m$ as follows:

$$y = y^p + A\Delta u + d \tag{A.11}$$

The objective of the MPC controller is to reduce the deviations of these output predictions from the desired set points. This research considers the QDMC algorithm (Garcia and Morshedi, 1986), which for MIMO systems can be formulated as the following quadratic programming (QP) problem.

$$\underset{\Delta u}{Min}\ J = \frac{1}{2}\cdot\left(\left\|y - y^{sp}\right\|_Q^2 + \left\|\Delta u\right\|_R^2\right) \tag{A.12}$$

$$s.t. \qquad y = A\Delta u + y^P + d \tag{A.12a}$$

Where $A \in \Re^{(p\cdot P)\times(m\cdot M)}$

$\Delta u \in \Re^{(m\cdot M)}$ $\qquad y, y^{sp}, y^P, d, \in \Re^{(p\cdot P)}$

$Q \in \Re^{(p\cdot P)\times(p\cdot P)}$ $\qquad R \in \Re^{(m\cdot M)\times(m\cdot M)}$

The control moves can be computed using the linear least square solution.

$$\Delta u = \left(A^T Q\ A + R\right)^{-1} A^T Q\ (-e) \tag{A.13}$$

Where A is the full dynamic matrix and vector $e = y^P - y^{sp} + d$ contains the feedback information. The tuning parameters Q and R are discussed later in this appendix.

Finally only the first element of each input variable, $\Delta u_{f,(1)}$ for $f = 1,\cdots,M$ is implemented and the entire procedure is repeated.

A.3.2 Constrained Model Predictive Control

One of the main strengths of MPC is its systematic way to handle constraints. In this section input and output constraints are introduced, and the resulting QP optimization problem is described.

First consider putting a constraint on the size of the control moves.

$$\left|\Delta u_{k+1}\right| \leq \Delta u^{\max} \tag{A.14}$$

These constraints preclude having severe control moves in order to bring the system back to the set point. Another approach typically employed in practice is to add some additional penalty terms to the objective function. This is called the move suppression factor, referred in thesis with parameter R, and it basically serve the dual purpose of suppressing aggressive control action and conditioning the system matrix prior to inversion (Shridhar and Cooper, 1998).

Another type of constraints on input variables account for the limits on the upper and lower values achievable by u_k. These constraints can be expressed as follows:

$$u_{k+i} = u_{k-1} + \sum_{j=0}^{i} \Delta u_{k+j} \tag{A.15}$$
$$u^{\min} \le u_{k+i} \le u^{\max} \qquad for\ i = 0,1,2,\cdots,m$$

These constraints contain the entire input horizon m.

Finally, the output constrains are incorporated by considering the effect of the control moves on the future output values.

$$y^{\min} \le y \le y^{\max}$$

Or using the model. (A.16)

$$y^{\min} \le A\Delta u + y^{p} + d \le y^{\max}$$

Adding this constrains to the input constraints may result in situations where there is no feasible solution to the QP problem. Softening the output constraints avoids this feasibility problem. Basically a nonnegative slack variable, $z \ge 0$ is created and added to the output constraint.

$$y^{\min} - z \le A\Delta u + y^{p} + d \le y^{\max} + z \tag{A.17}$$

A corresponding weighting factor, Z is added to the objective function. The resulting constrained MPC problem is the following:

$$\underset{\Delta u, z}{Min}\ J = \frac{1}{2} \cdot \left(\left\| y - y^{sp} \right\|_{Q}^{2} + \left\| \Delta u \right\|_{R}^{2} + \left\| z \right\|_{Z}^{2} \right) \tag{A.18a}$$

$$s.t. \quad y = A\Delta u + y^{p} + d \tag{A.18b}$$

$$y^{\min} - z \le y \le y^{\max} + z \tag{A.18c}$$

$$u^{\min} \le u \le u^{\max} \tag{A.18d}$$

$$\Delta u^{\min} \le \Delta u \le \Delta u^{\max} \tag{A.18e}$$

$$z \ge 0 \tag{A.18f}$$

The controller continuously minimizes the output constraint violations by minimizing the slack variable. The weighting matrices Q, R and Z can be chosen to achieve a desired degree of control over the constraint violation. In this work this weighting matrices have the following characteristics:

- Diagonal
- Same weight per variable throughout horizon
- Nonnegative values.

A.3.3 Tuning Parameters

This section presents a brief description of some of the tuning parameters used in the MPC algorithm. Some of these guidelines are taken from Marlin, (2000) and are followed throughout this thesis.

Sample time, Δt, and prediction horizon, p. The output horizon should be as long as the closed-loop settling time of the process, to guarantee that the process has reached steady-state at the end of the horizon. And the sample time should be a small fraction of the closed-loop settling time of the process.

The **input horizon,** m on the other hand should be kept small, typically one-quarter to one-third the length of the output horizon. Garcia and Morari (1982) showed that keeping $p >> m$ enhances the stability of the system.

As mentioned above the **move suppression factors**, R represent the relative importance of adjustments in each process input. These parameters are used to control the dynamic behavior of the system. The move suppression factors are also used to introduce a degree of robustness into the controller.

Finally the **weightings of the process outputs,** Q represent the relative importance of each output deviation from its set point.

Appendix B. Block Relative Gain (BRG)

The Block Relative Gain (BRG) generalizes the concept of RGA to handle multivariable block structures (Manousiouthakis, et al., 1986); therefore, it is a useful method for finding suitable pairings for block decentralized control. The BRG measures the interaction among multivariable controllers and in this appendix is explained using two blocks, although it can be generalized to any number of blocks.

$$\begin{pmatrix} y_1(s) \\ y_2(s) \end{pmatrix} = \left(\begin{array}{c|c} G_{11}(s) & G_{12}(s) \\ \hline G_{21}(s) & G_{22}(s) \end{array} \right) \begin{pmatrix} u_1(s) \\ u_2(s) \end{pmatrix} = G(s) \begin{pmatrix} u_1(s) \\ u_2(s) \end{pmatrix} \tag{B.1}$$

Where y_i and u_i are vectors of output and input variables, respectively, with dimension m_i.

The BRG for variable pairing (y_1, u_1) is defined as the ratio of the open-loop block gain matrix and the apparent gain matrix in the same loop when all other control loops are closed.

$$[A_B(s)]_{11} = G_{11}(s) \cdot \left[G^{-1}(s)\right]_{11} \tag{B.2}$$

Where $G_{11}(s)$ is the $m_1 x m_1$ transfer function relating the first m_1 inputs and outputs of $G(s)$ and $\left[G^{-1}(s)\right]_{11}$ is the corresponding block of $\left[G^{-1}(s)\right]$.

An alternative way to define the BRG is the following (Kariwala et al, 2003). Consider the LTI process $y(s) = G u(s) + d(s)$ to be partitioned such that $G_{ij}(s)$ is a $m_i x m_j$ transfer function matrix.

$$y_1(s) = G_{11}(s)u_1(s) + G_{12}(s)u_2(s) + d_1(s) \tag{B.3}$$

$$y_2(s) = G_{21}(s)u_1(s) + G_{22}(s)u_2(s) + d_2(s)$$

When (y_2, u_2) is perfectly controlled and $d(s) \approx 0$, at steady state, y_1 and u_1 are related through the Schur complement of G_{22}.

$$y_1 = \overline{G}_{11} u_1 \tag{B.4}$$

$$\overline{G}_{11} = G_{11} - G_{12} G_{22}^{-1} G_{21}$$

Now, The **steady-state** BRG between y_1 and u_1 can be defined as follows.

$$[\Lambda_B]_{11} = G_{11} \cdot \overline{G}_{11}^{-1} \tag{B.5}$$

BRG has some algebraic properties (Manousiouthakis, et al., 1986):

- Any permutation of rows and columns in the process open-loop process gain matri
- x $G(0)$ results in the same permutation in the BRG.
- BRG is independent of input scaling but dependent on output scaling. However, the diagonal elements of BRG are invariant under input and output scaling.
- The values of the diagonal elements of BRG are equal to the summation of all the relative gain values within the same rows.

Similar to RGA, BRG has rigorous relation with closed-loop properties such as

- Stability - Choosing a multivariable **diagonal** controller with negative determinant of BRG will cause at least one of three undesired situations: the multivariable control system by itself is unstable, the whole closed-loop system is unstable or the closed-loop system without the multivariable controller is unstable. Therefore, the general loop pairing guideline is to choose multivariable controller with positive determinant of BRG.
- Robustness - The spectral radius of any BRG associated with the system is the lower bound of Euclidean condition number of the system. In general, a control system with large maximum singular value of BRG is difficult to control.

- Integrity - Selecting control with positive determinant of BRG is a necessary condition for Integral Controllability with Integrity for block centralized structure (Chiu and Arkun, 1990).

Appendix C. Nominal Stability – A Two by Two Case.

In Chapter 4, the stability analysis of a "single-input horizon" *two by two* D-MPC controller was shown to be nominally unstable for negative BRG (RGA) for some values of the tuning constants. No rigorous analysis shows that this behaviour occurs for more complex D-MPC systems. However, we have encountered nominally unstable controllers, and this appendix documents some results obtained by numerical experimentation.

In this appendix the D-MPC control system in Case Study I (two by two distillation column) is analyzed for nominal stability. The analysis considers a negative RGA configuration, where the interactions are dominant. The tuning parameters considered for this analysis are the same used in the simulations and with exception of the move suppression factor (R), which is modified to analyze the regions of nominal stability. Figure C.1 shows the results of the analysis where three main regions are detected.

The methodology used for the analysis was basically trial and error, where (1) a set of move suppression factors is selected, (2) then a closed loop simulation is performed and (3) the nominal stability is verified using the closed loop stability expression developed in Chapter 4.

An important result is that the stable region is bounded between two different unstable zones. Therefore, we can state that when starting within the stable region, increasing or decreasing the values of R will produce an unstable controller. Figure C.2 contains the small section referred Figure C.1 and it shows an interesting behaviour.

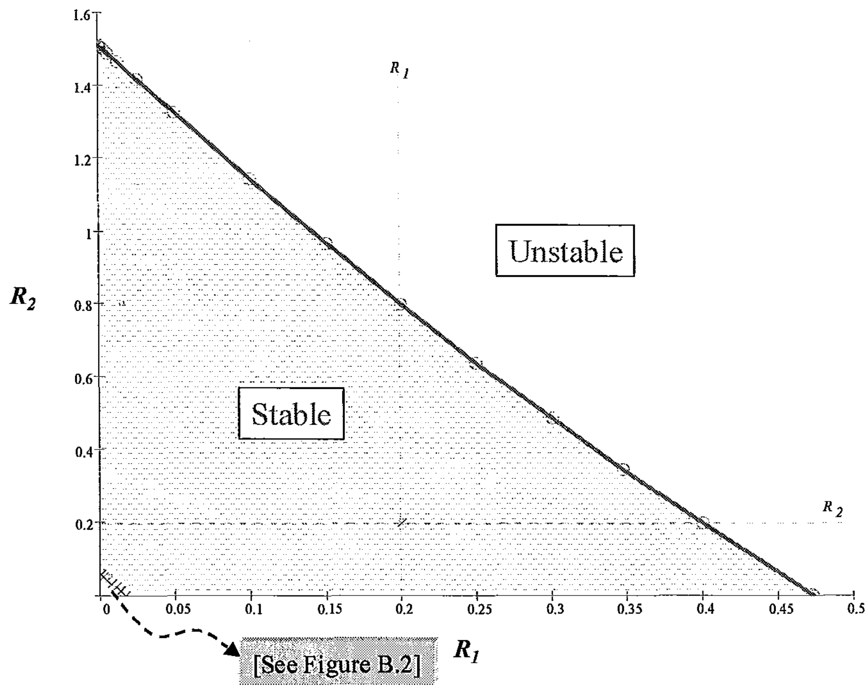


Figure C.1 Distillation Column - Stability Regions (Negative RGA)

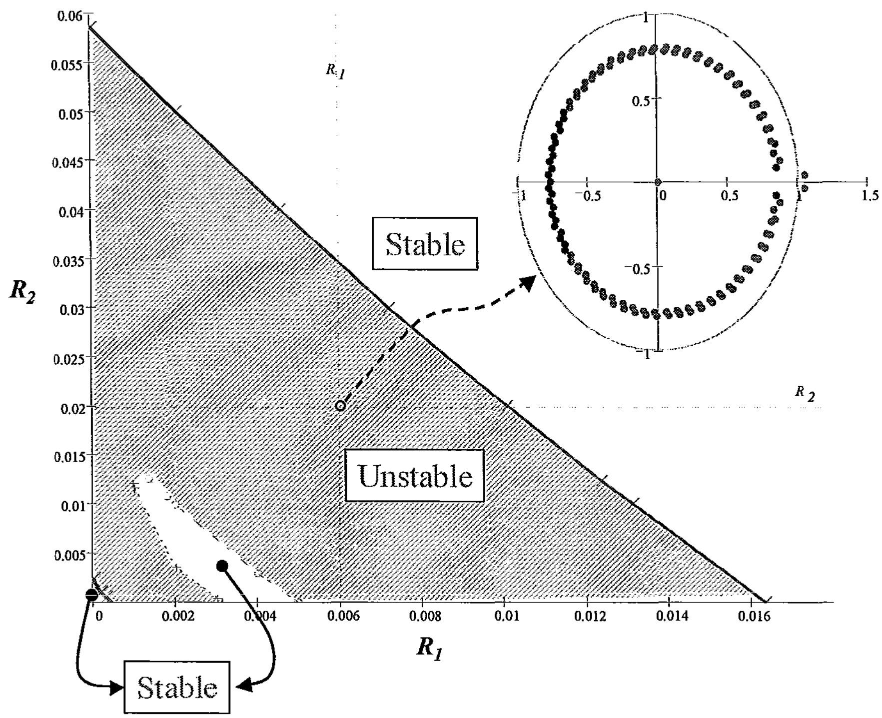


Figure C.2 Distillation Column - Stability Regions (Negative RGA)

In Figure C.2, it can be observed that the unstable region is surrounded between two different stable regions. Even more surprising is the fact that there is a small stable region within this unstable region. Finally the smallest region (stable) contains the origin $(R = 0)$ which follows because the process in open-loop stable.

Finally, it is important to remark that a considerable number of tuning combinations were evaluated to obtain the results illustrated in the figures; however, it is acknowledged that there could be other regions.

Appendix D. Distillation Column with Model Mismatch.

The cases in the body of the thesis reported results for D-MPC control systems without mismatch. Therefore, this appendix presents the distillation column case study with model-mismatch. The different cases consider situations where the feed flow rate has either increased or decreased by a certain percentage. The process dynamic parameters for the two mismatch cases and the nominal case are given in Table D.1

Table D.1 Process Dynamic Parameters

Parameter	Description	Case 1 (20% Feed Increase)	Nominal Case (Model in Controller)	Case 2 (20% Feed Decrease)
Kp	Steady State Gain, $(\text{kgmol/min})^{-1}$	$\begin{pmatrix} 0.0623 & -0.0556 \\ 0.0978 & -0.1044 \end{pmatrix}$	$\begin{pmatrix} 0.0747 & -0.0667 \\ 0.1173 & -0.1253 \end{pmatrix}$	$\begin{pmatrix} 0.0934 & -0.0834 \\ 0.1466 & -0.1566 \end{pmatrix}$
τ	Time constant, (min)	$\begin{pmatrix} 10 & 12.5 \\ 9.8 & 8.5 \end{pmatrix}$	$\begin{pmatrix} 12 & 15 \\ 11.75 & 10.2 \end{pmatrix}$	$\begin{pmatrix} 15 & 18.75 \\ 14.68 & 12.75 \end{pmatrix}$
θ	Dead time, (min)	$\begin{pmatrix} 2.5 & 1.67 \\ 2.75 & 1.67 \end{pmatrix}$	$\begin{pmatrix} 3 & 2 \\ 3.3 & 2 \end{pmatrix}.$	$\begin{pmatrix} 3.75 & 2.5 \\ 4.13 & 2.5 \end{pmatrix}$

The case studies presented in this appendix have the objective to demonstrate to a certain extent that the D-MPC controller provides sufficient robustness to handle model uncertainty. The tuning for these cases is the same as reported in Section 5.3 for the D-MPC controller.

The first case considers a situation where the feed flowrate has increased by 20%, let's note that because the levels on trays and accumulation vessels are assumed not to change significantly the dynamics of the process becomes faster. The process is simulated for set point changes in X_D and X_B. Input constraints and noise measurement are also considered.

Figure D.1 illustrates the closed loop simulation using a positive RGA configuration and Figure D.2 shows the simulation using a negative RGA configuration. The results are as expected where the D-MPC produces offset only in the loop (block) in which the manipulated variables saturates.

Te second case considers a feed flowrate decrease of 20%. Figure D.3 and Figure D.4 show the closed loop simulations for positive and negative RGA configurations respectively.

In all cases, the dynamic behavior is satisfactory. The dynamic responses are stable and do not experience undesirable behavior (excessive oscillations or overshoot). While these results do not provide guarantees for robust performance, they demonstrate that the D-MPC controller has promise for realistic applications with mismatch.

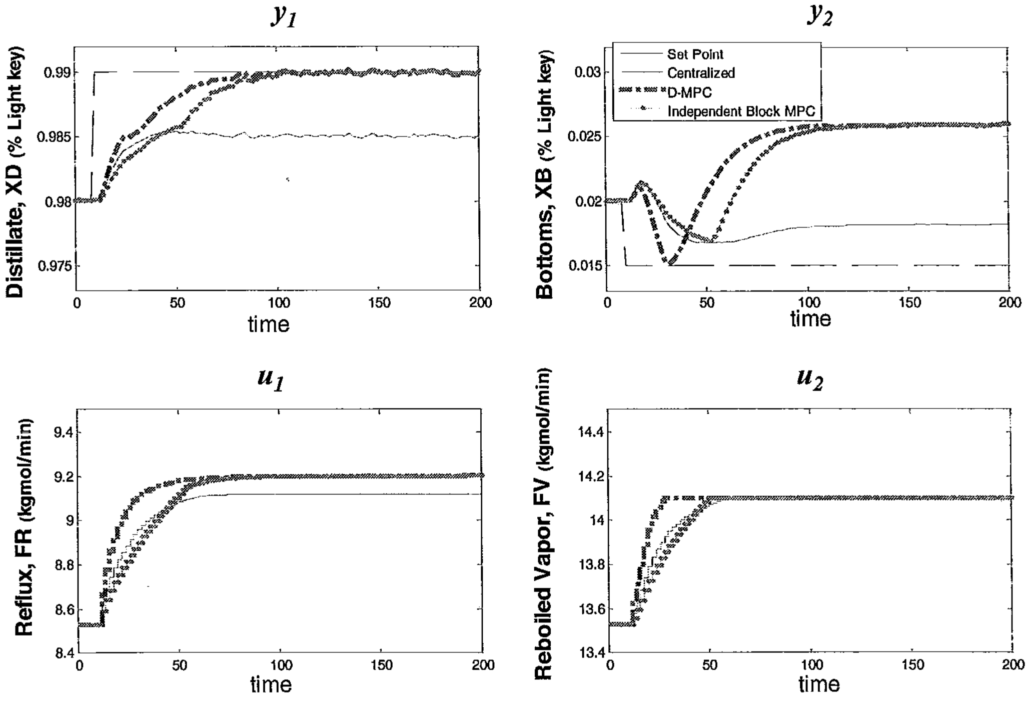


Figure D.1 Constrained Control – 20% Feed Flowrate Increase (Positive RGA)

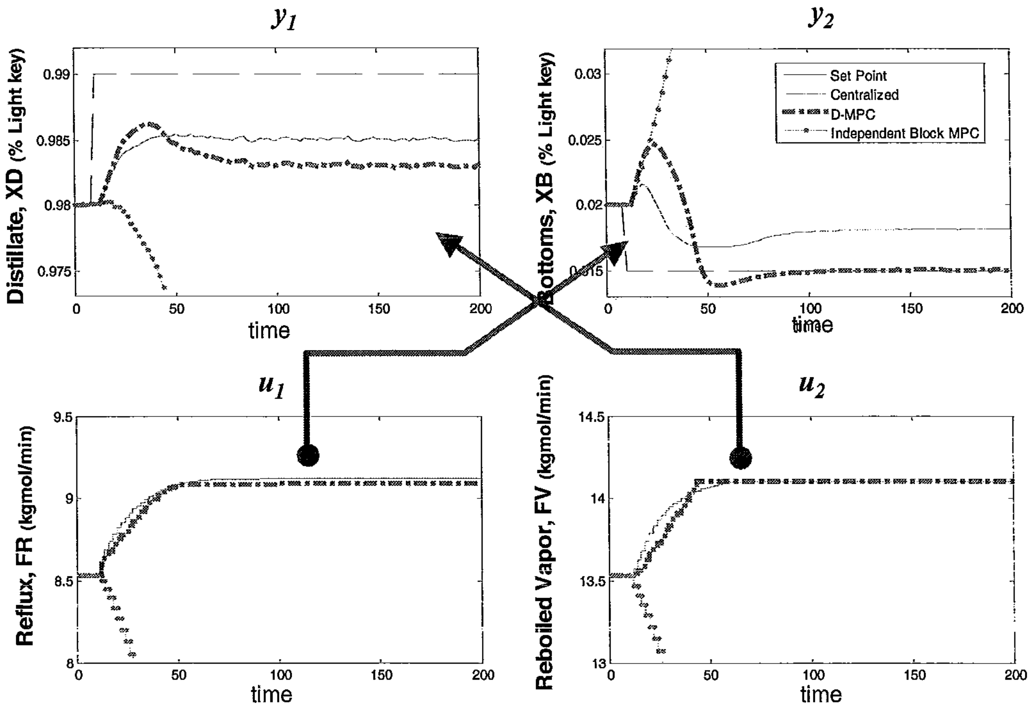


Figure D.2 Constrained Control – 20% Feed Flowrate Increase (Negative RGA)

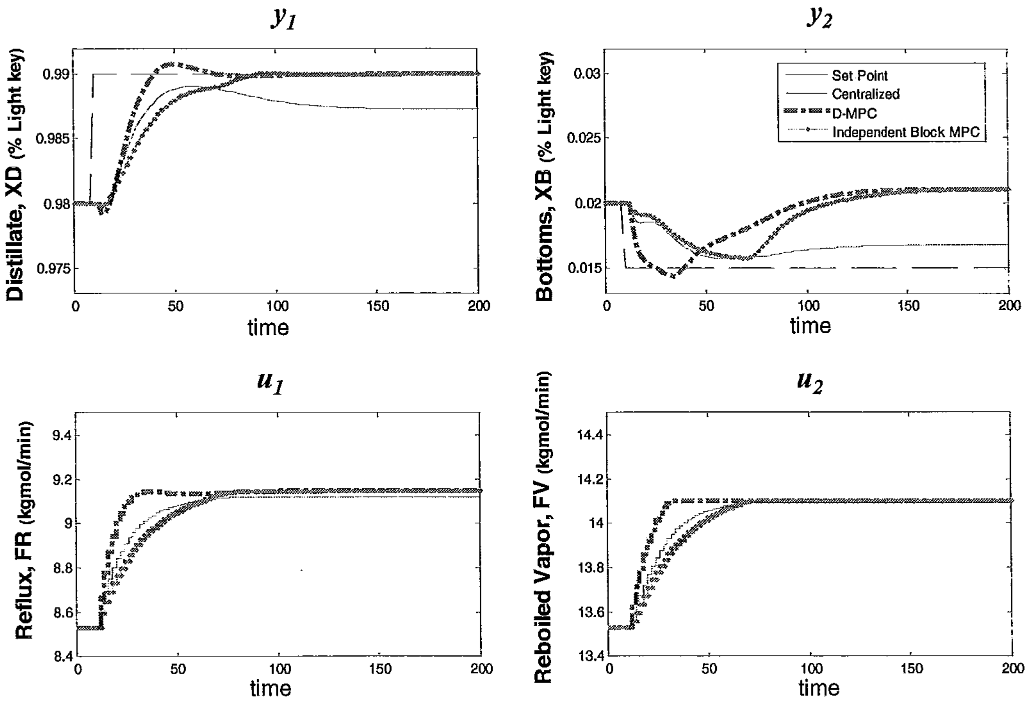


Figure D.3 Constrained Control – 20% Feed Flowrate Decrease (Positive RGA)

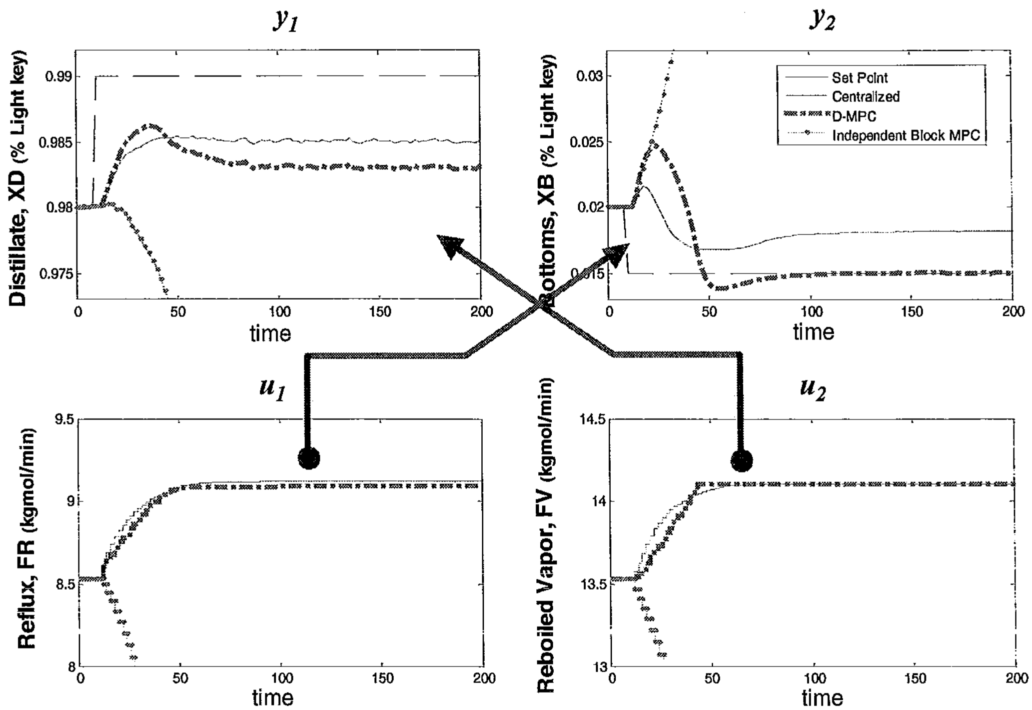


Figure D.4 Constrained Control – 20% Feed Flowrate Decrease (Negative RGA)

Appendix E. Software Implementation

This appendix describes the basic structure of the software developed for the implementation of the D-MPC controller. Figure E.1 presents a schematic representation of the software structure for a given case study.

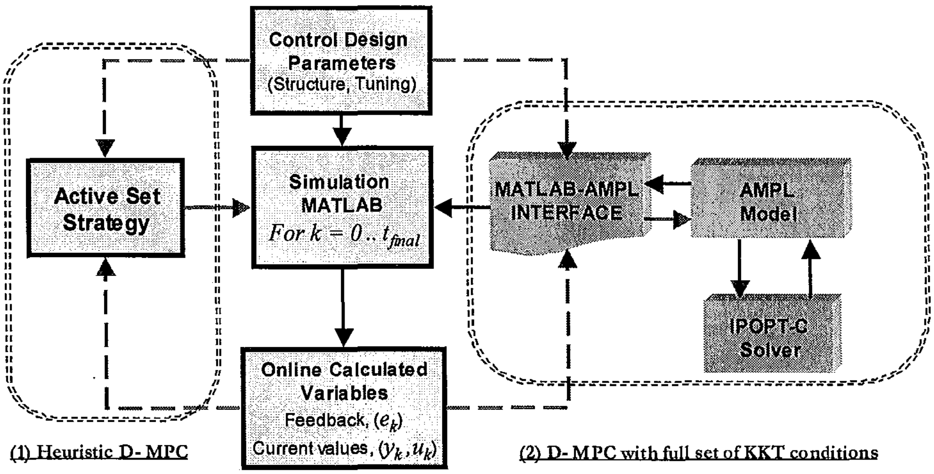


Figure E.1 Software Structure for a Given Process

The core of the simulation is programmed in MATLAB, but the controller calculations could be performed of two different forms depending on the selected D-MPC strategy.

(1) If the Heuristic D-MPC is selected an active set strategy programmed in MATLAB is executed. The basic active set algorithm is described in Section 3.4.

(2) For the D-MPC that uses the full set of KKT conditions the software requires an interface program that links MATLAB and the modelling language AMPL, this in turn calls for the IPOPT-C solver and then return the control calculations to the MATLAB simulator. The MATLAB-AMPL interface developed for this research basically transforms a set of MATLAB variables into a set of data files (*.dat) that are stored and then accessed from a hard disk.