SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

SEVENTH FRAMEWORK PROGRAMME

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC

HD–MPC

Deliverable Number:	D3.1.4
Deliverable Type:	Report
Contractual Date of Delivery:	September 1, 2011
Actual Date of Delivery:	**August 31, 2011**
Title of Deliverable:	**Reports on the evaluation results, including economical potential and suggestions for real-life applications**
Dissemination level:	Public
Workpackage contributing to the Deliverable:	WP3
WP Leader:	Wolfgang Marquardt
Partners:	RWTH, USE, POLIMI, TUD, UNC, SUPELEC
Author(s):	H. Scheu, A. Núñez, B. De Schutter, D. Sáez, C. E. Cortés, M. D. Doan, T. Keviczky, F. Valencia, J. Espinosa

Table of contents

Project co-ordinator

Name: Bart De Schutter
Address: Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 Delft, The Netherlands
Phone Number: +31-15-2785113
Fax Number: +31-15-2786679
E-mail: b.deschutter@tudelft.nl

Project web site: `http://www.ict-hd-mpc.eu`

Executive Summary

This report presents further advances in the development of hierarchical and distributed model predictive control (MPC) methods. In particular, we present a hierarchical approach for a Dynamic Pickup and Delivery Problem. The hierarchical multilayer structure of the system is used to decompose the optimization problem, which is big and NP-hard, into smaller but more tractable subproblems. Each proposed layer represents the viewpoint of different decision-makers. In one of those layers, the dispatcher routes the vehicles when a new request appears, and minimizes user and operator costs. As those two components are usually aimed at opposite goals, the problem in this layer is formulated and solved through multiobjective model predictive control.

Then a game-theoretic approach is presented for distributed model predictive control (DMPC). The DMPC problem is reformulated and analyzed as an *n*-person bargaining game based on the concepts presented by John Nash. The *n*-person bargaining game involves *n* individuals that can collaborate for mutual benefit. The individuals communicate with each other in order to (jointly) decide which strategy is the best for each individual, based on the profit received under cooperative behavior.

The third and last method presented in this report is a hierarchical and distributed approach. The proposed scheme facilitates the implementation of MPC without building a powerful centralized controller, which is often impractical for large-scale systems. The proposed method is applicable to a large class of interconnected systems where there can be couplings in both dynamics and constraints between the subsystems. The hierarchical MPC controller is able to generate a primal feasible solution within a finite number of iterations, using primal averaging and a constraint tightening approach.

All these approaches are first briefly presented in the synopsis chapter, while a full presentation can be found in the subsequent chapters. Additionally, the synopsis chapter presents the economical potential and suggestions for real-life applications.

Chapter 1

Synopsis

This synopsis chapter first summarizes further results developed for hierarchical and distributed model predictive control, namely:

- Hierarchical Multiobjective Model Predictive Control Applied to a Dynamic Pickup and Delivery Problem.
- Game Theory Based Formulation of Distributed Model Predictive Control.
- A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints.

Then, the synopsis is concluded with a section summarizing the economical potential of hierarchical and distributed MPC methods and suggestions for real-life applications.

1.1 Hierarchical multiobjective model predictive control applied to a dynamic pickup and delivery problem

The advances in Hierarchical and Distributed Model Predictive Control (HD-MPC) during the last decade have made this framework very attractive for dealing with problems associated with the management of real-time operations involved in complex operational processes. In this sense, the problems that arise in the operation of transport systems have become of real interest for applying not only the methodology, principles and modeling techniques behind HD-MPC, but also several of the new solution algorithms that have proved to be efficient in the context of HD-MPC applications.

The decisions about operational policies, were mostly conducted relying on static optimization methods to make decisions. These static methods were used even though the dynamism in the operation of most transport systems is nowadays widely recognized as part of their natural interaction with the demand and infrastructure. The reasons for using static scenarios and models for such long time were mainly due to computational constraints, lack of efficient algorithms and proper technology, etc.

On the contrary, in the last fifteen years, researchers have intensively worked to deal with dynamic transport modeling and control problems, which has changed completely the way to conceive the algorithms and policies used for planning the operation of the transport systems involved. Issues such as data management, computational performance, future conditions prediction and real-time decisions became relevant in the conception of operational schemes for several types of transport systems.

By looking into most of the specialized literature regarding such dynamic methods and algorithms, the real-scale transport problems are commonly treated through heuristic methods, which does not seem to be a bad option even the most important operational decisions that are conditioned by the algorithms' solutions normally must be made in real-time. In these cases, it is worth regularly re-evaluating the last policy applied in order to reach a better performance in the medium to long-term time-scale, like in a rolling horizon fashion. In fact, the use of static approaches adapted to solve dynamic problems can considerably underestimate the potential benefits of certain dynamically derived operational policies for both private operators and for societal end-users (users of transport systems).

Some of the dynamic transport problems that naturally fit to HD-MPC, with the dynamic features of the most common transport schemes are i) dynamic vehicle routing problems (of passengers, loads), ii) real-time operations of traditional (fixed route) public transport systems (buses, train), iii) real time traffic control (urban and in highways), etc.

Indeed, in these applications the description of the future behavior associated with the operational processes generate highly non-linear model-based predictive control formulations containing a combination of integer and continuous variables, soft and hard constraints, etc. Therefore, given the complexity of those systems, it will be important to end up with concise and efficient model descriptions along with a proper predictive objective, to make sure that the HD-MPC methods are applicable to the real-time settings of the transport systems analyzed.

In Chapter 2 [1], a hierarchical multiobjective model based predictive control approach is presented for solving a dynamic pickup and delivery problem. The hierarchical multilayer structure of the system is used to decompose the optimization problem, which is big and NP-hard, into smaller but more tractable subproblems. Each proposed layer represents the viewpoint of different decision-makers. In one of those layers, the dispatcher routes the vehicles when a new request appears, and minimizes user and operator costs. As those two components are usually aimed at opposite goals, the problem in this layer is formulated and solved through multiobjective model predictive control. The dispatcher participates in the dynamic routing decisions by expressing his/her preferences in a progressively interactive way, seeking the best trade-off solution at each instant among the Pareto optimal set. An illustrative experiment of the new approach through simulation of the process is presented to show the potential benefits in the operator cost and in the quality of service perceived by the users.

1.2 Game theory based formulation of distributed model predictive control

Game theory is a branch of applied mathematics used in social sciences, economics, biology (particularly evolutionary biology and ecology), engineering, political science, international relations, computer science, and philosophy. Game theory attempts to capture behaviors in strategic situations, or games where the outcome of a player is function not only of his choices but also depends on the choices of others [2]. While initially developed to analyze competitions in which one individual does better at another's expense, it has been expanded to treat a wide class of interactions, which are classified according to several criteria. Today, "game theory is a sort of 'unified field' theory for the rational side of social science, where 'social' is interpreted broadly, to include human as well as non-human players (computers, animals, plants)"[3]. Thus, game theory arises as an alternative to formulate and characterize the distributed model predictive control (DMPC) problem.

In Chapter 3, the DMPC problem is reformulated and analyzed as a n-person bargaining game based on the concepts presented by John Nash in [4, 5, 6] about such games. The n-person bargaining game involves n individuals that can collaborate for mutual benefit. The individuals communicate with each other in order to (jointly) decide which strategy is the best for each individual, based on the profit received under cooperative behavior [4]. So, in the proposed formulation, each subsystem is able to decide whether to cooperate or not with the other subsystems depending on the benefit received by the subsystem from the cooperative behavior. The selection of the bargaining approach was made because its main insight is focusing on others, i.e., to assess your added value, you have to ask not what other players can bring to you but what you can bring to other players [7].

For analyzing the DMPC problem as a *n*-person bargaining game the axiomatic bargaining theory is used. Since this theory is formulated for static games, some axioms and concepts have been redefined. Moreover, the concept of discrete-time dynamic bargaining game is introduced. Based on the new concepts, two cases of distributed model predictive control are analyzed: the symmetric and the nonsymmetric cases (conditions for the symmetry and nonsymmetry of the game associated with the DMPC problem are established). For both cases the outcome of the game is characterized, i.e., the properties of the DMPC formulated as a bargaining game are discussed.

In addition, a negotiation model for implementing a distributed solution of both symmetric and nonsymmetric DMPC games is presented. This algorithm is based on the transformation of the bargaining game in an equivalent noncooperative game, and solve the equivalent noncooperative game. The transformation allowed to reduce the computational burden associated with the solution of the DMPC problem because it is not required an iterative procedure for jointly compute the optimal control action applied to each subsystem. Also, the difference between the proposed algorithm and the other algorithms for DMPC (specifically the DMPC algorithms based on Lagrange multipliers) are discussed, and the conditions for the convergence and the stability of the proposed DMPC scheme are established.

Finally, the quadruple tank process is used to illustrate a symmetric case, and a hydro-power valley is used to present a nonsymmetric case.

1.3 A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints

Chapter 4 [8] presents a new approach in designing model predictive controllers, based on hierarchical and distributed MPC architecture. The proposed scheme facilitates the implementation of MPC without building a powerful centralized controller, which is often impractical for large-scale systems. The proposed method is applicable to a large class of interconnected systems where there can be couplings in both dynamics and constraints between the subsystems. The hierarchical MPC controller is able to generate a primal feasible solution within a finite number of iterations, using primal averaging and a constraint tightening approach. The primal update is performed in a distributed way and does not require exact solutions, while the dual problem uses an approximate subgradient method. Stability of the scheme is established using bounded suboptimality.

We consider M interconnected subsystems with coupled dynamics, the centralized discrete-time state-space model is given by:

$$x_{k+1} = Ax_k + Bu_k$$

Let N be the receding horizon. In the MPC problem at time step t, the subsystems need to respect the operational constraints:

$$\begin{aligned}
&x_k \in \mathscr{X}, k = t+1, \ldots, t+N-1 \\
&x_{t+N} \in \mathscr{X}_\mathrm{f} \subset \mathscr{X} \\
&u_k \in \mathscr{U}, k = t, \ldots, t+N-1 \\
&u_k^i \in \Omega_i, i = 1, \ldots, M, \quad k = t, \ldots, t+N-1
\end{aligned}$$

where the coupled constraint sets $\mathscr{U}$, $\mathscr{X}$ and $\mathscr{X}_{\mathrm{f}}$ are polytopes and have nonempty interiors, and each local constraint set Ω_i is a hyperbox.

The MPC problem at time step t is formed using a convex quadratic cost function. After eliminating the state variables, we need to solve the following optimization problem at time step t:

$$\begin{aligned} f_t^* = \min_{\mathbf{u}} \quad & f(\mathbf{u}, x_t) \\ \text{s.t.} \quad & g(\mathbf{u}, x_t) \leq 0 \\ & \mathbf{u} \in \mathbf{\Omega} \end{aligned} \tag{1.1}$$

where f and $g = [g_1, \dots, g_m]^T$ are convex functions, and $\mathbf{\Omega} = \prod_{i=1}^{M} \mathbf{\Omega}_i$ with each $\mathbf{\Omega}_i = \prod_{k=0}^{N-1} \Omega_i$ is a hyperbox.

Suppose that at each time step t, we have a Slater vector $\bar{\mathbf{u}}_t$ which satisfies the strict inequality $g(\mathbf{u}, x_t) < 0$, and we can quantify the difference between the costs associated with the previous step and the Slater vector as:

$$f(\mathbf{u}_{t-1}, x_{t-1}) - f(\bar{\mathbf{u}}_t, x_t) > \Delta_t$$

Our objective is to generate a feasible solution for problem (1.1) using a method that is favorable for distributed computation. The main idea is to use dual decomposition for the tightened problem of (1.1), such that after a finite number of iterations the constraint violations in the tightened problem will be less than the difference between the tightened and the original constraints. Thus, even after a finite number of iterations, we will obtain a primal feasible solution for the original MPC optimization problem. Moreover, the suboptimality would be less than Δ_t, so that the cost function decreases, acting as the candidate function for proving Lyapunov stability.

The novel control technique is based on a two-level hierarchical and distributed optimization algorithm, which is a nested procedure in which the outer loop is the approximate subgradient method for the dual problem and the inner loop is the Jacobi distributed optimization method for the primal problem. Most of the computations are carried out by the local controllers in a distributed fashion, while the coordinator is in charge of computing the common parameters: the tightening offset c_t, the suboptimality ε_t, the step size α_t of the subgradient iteration, the number of outer-loop iterations $\bar{k}_t$ and the number of inner-loop iterations $\bar{p}_k$.

We show that the average of the primal update series is a feasible solution and lead to closed-loop MPC stability with the following parameters to be defined *a priori*:

$$\begin{aligned} & 0 < c_t < \min_{j=1,\dots,m} \{-g_j(\bar{\mathbf{u}}_t, x_t)\} \\ & \alpha_t = \frac{\Delta_t}{L_t'^2} \\ & \varepsilon_t = \frac{\Delta_t}{2} \\ & \bar{k}_t = \left\lceil \frac{1}{\alpha_t c_t} \left(\frac{3}{\gamma_t} f(\bar{\mathbf{u}}_t, x_t) + \frac{\alpha_t L_t'^2}{2\gamma_t} + \alpha_t L_t' \right) \right\rceil \\ & \bar{p}_k = \left\lceil \log_\phi \frac{\varepsilon_t}{\Lambda M \max_i D_i} \right\rceil \end{aligned}$$

where $\lceil \cdot \rceil$ is the ceiling operator which gives the closest integer equal to or above a real value, M is the number of subsystems, $L_t', \gamma_t, \phi, \Lambda$, and D_i are the constants that are associated with the properties of the functions f and g.

1.4 Economical potential and suggestions for real-life applications

Estimation of the economical potential of the new hierarchical and distributed methods is difficult task. However, there are some results in literature, which can deal as a basis for this estimation. It is obvious that a good centralized solution of an optimal control problem provides an upper bound for the best control performance, which can be achieved by hierarchical and distributed model predictive control methods. A large scale study for the load change of a chemical plant has been provided by Hartwich and Marquardt [9]. They report a profit gain of up to 8.6 % using an optimal controller during transit. However, the process considered involves more than 12.000 variables, so the solution of the optimal control problem can only be accomplished in several days. Hence, a real-time application is not possible.

In another case study, which is still large-scale Würth et al. [10] demonstrate the capabilities of modern optimal control methods. In particular, they compare single-layer optimal control and hierarchical multi-layer control for a continuous polymerization process, which involves approximately 2000 nonlinear equations. In total, 4 different control scenarios are presented. The first controller is a single-layer NMPC, where computational delay is neglected. The results for this controller are considered as the optimum achievable. However, neglecting the computational delay is an unrealistic assumption, since the nonlinear optimal control problem cannot be solved in real-time. If computational delay is considered, the controller performance dramatically reduced: The economic cost function to be minimized doubles, while the number of constraint violations rises by a factor of 10. But, using a hierarchical nonlinear model predictive control scheme, the controller performance of the reference controller can be almost retained. This clearly demonstrates the advantages of hierarchical MPC, at least in a large-scale simulation case study. Though, this case study also shows, that there are limitations in the size of a system, which can be dealt with by a monolithic or hierarchical controller, as the computational delay can not be neglected. For systems with faster time-constants, the situation gets even worse.

Here, the use of distributed model predictive control methods can be the solution to broaden the class of systems, for which optimal control can be realized. This is especially important, when nonlinear systems are considered. However, many of the methods in distributed model predictive control are still not mature, normally they only exist for linear systems, as those presented in [11]: This article presents a comparative analysis of different model predictive controllers. While centralized controllers serve as a reference, different distributed and fully decentralized controllers are studied in terms of various performance criteria. An important message is that distributed MPC solutions can achieve the same controller performance as a centralized controller, while the same cannot be achieved using decentralized MPC. The evaluations are conducted for a real plant, namely the well-known quadruple-tank process suggested by Johansson [12]. Although this is a rather small-scale example, the evaluation is one of the first studies of distributed MPC on a real plant, while the majority of contributions are based on simulations.

Generally, a main benefit from using distributed methods compared to decentralized methods can be seen in case a decentralized controller leads to unstable system behavior of a MIMO plant. On the one hand, if the use of decentralized controller is to be used, a way to tackle the problem might be to change the controller tuning such that the system performance is reduced. As a result, reference tracking and disturbance rejection gets worse. Then, the plant cannot be operated as close to the system bounds, as one might want to do this, hence, one the plant is not operated at the economical optimum. On the other hand, the use of a centralized controller leads to more challenging computational burdens, resulting in bigger sampling times and longer computational delays, which again reduces the controller performance and forces the operator to operate the plant apart from its economical optimum.

Hence, if a distributed controller guarantees stability and optimality and reduces computing time compared to a centralized controller, there is a continuous profit for the distributed over the centralized solution. Compared to a decentralized controller the profit gain depends on the coupling of the subsystems. For uncoupled subsystems, a completely decentralized controller still provides optimal performance, optimal disturbance rejection and optimal reference tracking. This might still be a good solution for very weakly coupled subsystems. But as the couplings get stronger, a distributed controller will definitely perform better than the decentralized controller. The stronger the couplings are, the stronger should be the benefit of using a distributed MPC over a decentralized MPC. However, for very strong coupling, e.g. consider a single CSTR with multiple inputs and outputs, the use of a distributed solution will most likely be worse than a centralized MPC. As a result, we want to stress that before deciding on the control method to be implemented, one should analyze the MIMO system using existing tools such as the relative gain array, in order to decide on whether to implement distributed MPC or not.

Finally, we want to give a coarse guess on a real plant, in particular a 1 GW power plant. However, as real numbers are hardly available, the following numbers are really coarse guesses in order to get a feeling on the real benefits of new MPC technology. We assume the plant is operating 8000 hours per year and a cost of 10 cent per kWh. Hence, the plant is producing a total of 8 TWh per year, which is equivalent to $8 \times 10^{12}\text{Wh}\frac{0.10\text{Euro}}{1\times 10^{3}\text{Wh}} = 800.000.000$. Even if we assume only an improvement in efficiency of the plant of 0.01 % due to implementation of a distributed or hierarchical MPC, costs can be reduces by about 80.000 Euro per year, for an assumed reduction of 0.1 %, it is already a saving of 800.000 Euro per year.

From, these numbers, it gets clear, that the effort of implementing an improved control concept, e.g. distributed or hierarchical MPC, for a large scale plant, will easily pay back the effort in a reasonable time. For smaller plants, the benefit will be seen, as soon as the same solution will be applicable on multiple systems. However, a necessary requirement for adaption in industrial plants is still the bring hierarchical and distributed MPC methods to a more mature status.

Bibliography

[1] A. Núñez, B. De Schutter, D. Sáez, and C. Cortés, "Hierarchical multiobjective model predictive control applied to a dynamic pickup and delivery problem," in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010)*, (Madeira Island, Portugal), pp. 1553–1558, Sept. 2010.

[2] R. B. Myerson, *Game theory: analysis of conflict*. Harvard University Press, 1991.

[3] R. J. Aumann, "Game theory, the new palgrave," *A Dictionary of Economics*, vol. 2, pp. 460–482, 1987.

[4] J. F. Nash, "Equilibrium points in N-persons games," *Proceedings of the National Academy of Sciences of United States of America*, vol. 36, no. 1, pp. 48–40, 1950.

[5] J. Nash, "The bargaining problem," *Econometrica*, vol. 18, no. 2, pp. 155–162, 1950.

[6] J. Nash, "Two-person cooperative games," *Econometrica*, vol. 21, no. 1, pp. 128–140, 1953.

[7] A. M. Brandenburger and B. J. Nalebuff, "The right game: Use the game theory to shape strategy," *Harvadr Bussines Review*, July-August 1995.

[8] M. D. Doan, T. Keviczky, and B. De Schutter, "A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints," in *Proceedings of the CDC and ECC*, 2011.

[9] A. Hartwich and W. Marquardt, "Dynamic optimization of the load change of a large-scale chemical plant by adaptive single shooting," *Computer & Chemical Engineering*, vol. 34, pp. 1873–1889, 2010.

[10] L. Würth, R. Hannemann, and W. Marquardt, "A two-layer architecture for economically optimal process control and operation," *Journal of Process Control*, vol. 21(3), pp. 311–321, 2011.

[11] I. Alvarado, D. Limon, D. M. de la Peña, J. Maestre, M. Ridao, H. Scheu, W. Marquardt, R. Negenborn, B. D. Schutter, F. Valencia, and J. Espinosa, "A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark," *Journal of Process Control*, vol. 21, no. 5, pp. 800 – 815, 2011. Special Issue on Hierarchical and Distributed Model Predictive Control.

[12] K. H. Johansson, "The quadruple-tank process," *IEEE Transactions on Control Systems Technology*, vol. 8, pp. 456 – 465, 2000.

Chapter 2

Hierarchical Multiobjective Model Predictive Control Applied to a Dynamic Pickup and Delivery Problem

The research of this chapter has been developed by Alfredo Núñez, Bart De Schutter, Doris Sáez, and Cristián E. Cortés. A. Núñez and B. De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands D. Sáez is with the Department of Electrical Engineering, Universidad de Chile, Av. Tupper 2007, Santiago, Chile. C.E. Cortés is with the Department of Civil Engineering, Universidad de Chile, Blanco Encalada 2002, Santiago, Chile.

2.1 Introduction

The dynamic pickup and delivery problem (DPDP) considers a set of online requests of service for passengers traveling from an origin (pickup) to a destination (delivery) served by a fleet of vehicles initially located at several depots [1], [2], [3]. The final output of such a problem is a set of routes for the fleet, which dynamically change over time and are required in real-time. The DPDP designed to operate dial-a-ride systems (DARS) has been intensely studied in the last decades [4], [5], [6], [7], among which the ADART system in Corpus Christi Texas, which is a distributed system for dynamic routing already implemented in real-life [8].

A well-defined DPDP should be based on an objective function that includes prediction of future demands and traffic conditions in current routing decisions, [9], [10], [11], [12]. In previous works we have proposed an analytical formulation for the DPDP as a model based predictive control (MPC) problem. The proposed global optimization problem was big and NP-hard, so, the use of evolutionary algorithms was considered. However, the global optimum solution in real-time instances was not reached due to the trade-off between computation time and accuracy in those algorithms. In this paper, we propose a new control structure for DPDP that does not only incorporate predictions, but also the inherent hierarchical multilayer and multiobjective structure of the DPDP.

Regarding hierarchical model based predictive control (HMPC), a very nice and comprehensive review can be found in [13]. The references within [13] represent the main contributions in the field. In a HMPC structure the local actions of the controllers are coordinated by an algorithm operating at a higher level. The higher layers determine general characteristics of the system and generate control variables which have a long-term effect on the plant. Those variables usually are obtained by a static optimization procedure and remain constant during a relative longer period of time. In the lower layers, control variables are determined by means of a higher rate optimization procedure (MPC) and their effects are local and short-term (see more details in [13], [14] and [15]). In this paper, we propose a hierarchical scheme with three layers for solving the DPDP, where each layer represents the viewpoint of different decision-makers. The communication and coordination issues in each layer are very important, not only because the information is received at different rates, but also because of some conflicts that could happen especially when their objectives are opposite. This is the case for DPDP, when we consider quality of service for users while minimizing operational costs.

In real implementations of DPDP the quality of service is very important. The authors in [16] conclude that most dial-a-ride studies are focused on the minimization of operational costs, and that it is necessary to develop more studies on user-policies. Then, it is reasonable that the objective function properly quantifies both the impact on the users' level of service affected by real-time routing decisions, as well as the effect on the associated extra operational costs. We must notice that these two dimensions represent opposite objectives and we will need to solve conflicts between them. The users want to obtain good service, implying more direct trips, resulting in lower vehicle occupancy rates and consequently, higher operational costs. More efficient routing policies from the operators' standpoint will reflect higher occupation rates, longer routes, and consequently, longer waiting and travel time for users. Thus, the question is how to properly balance both components in the objective function to make proper dispatching decisions. To guide the decision-maker in this line, in this paper we propose the use of multiobjective model based predictive control. The dispatcher must express his/her preferences (criterion) in a progressive way (interactively), seeking the best-compromise solution from the dynamic Pareto set. The performance of the system will be related with the criterion used.

Multiobjective optimization (MO) has been applied to a large number of static problems and also to vehicle routing problems [17], [18]. For a comprehensive review, the interested reader is referred to [19]. As far as we know, all the multiobjective applications in vehicle routing problems are evaluated in static scenarios, one of the aims of this paper being to contribute in the analysis of using MO in dynamic and stochastic environments. Among works and applications related to MO in MPC we can highlight [20], [21], and [22]. Almost all the works reported in this line prioritize, or use scalars methods by weighting the objective functions (a priori) turning the MO problem into a single-objective optimization. Those methods are too rigid in the sense that changes in the preference of the decision-maker cannot easily be considered. Then, we propose a suitable tool for dispatchers that allows to make decisions in a more transparent way.

The outline of the paper is as follows. In Section 2, the Hierarchical Multiobjective Model Predictive Control approach is presented. In Section 3 the DPDP, including the model and the objective functions are discussed. In Section 4, the scheme based on MPC for solving the DPDP by [12] is reformulated under the new approach. In Section 5 simulation results are shown and analyzed. Finally conclusions and future work are highlighted.

2.2 Hierarchical multiobjective model predictive control

2.2.1 Hierarchical model predictive control

In hierarchical multilayer systems, the system is divided into different functional layers, and the control structure consists of algorithms dealing with different components of the system, working at different temporal and spatial scales. This structure is useful to control plants characterized by significantly different dynamics and where the action of local controllers is coordinated by an algorithm operating at a higher level [13]. In process industry it is very common to design the overall control system according to hierarchical structures [24]. In some applications, and also in the one we propose in this paper, at a higher level a simpler and more abstract model is considered to predict the long-term behavior of the system and to compute the optimal plant operating conditions based on an economic criterion. At the lower level, a more accurate model is used to compute the current control actions by looking at a shorter time horizon.

Consider for example a hierarchical model based predictive controller with a given number of layers. The variables of the higher layer are denoted with superscript 1, in the next lower level the superscript is 2 and so on. The process modeled in layer s is given by the following non-linear discrete-time system:

$$x^s(k_s+1) = f_s(x^s(k_s), u^s(k_s), X^s(k_s)), \tag{2.1}$$

where $x^s(k_s) \in \mathbb{R}^{n_s}$ is the state vector, $u^s(k_s) \in \mathbb{R}^{m_s}$ is the input vector, $X^s(k_s)$ are states and inputs from higher layers that affect the dynamics of layer s, and $k_s \in \mathbb{N}$ denotes the time step in layer s. In the layer s, the following MPC problem is solved:

$$\begin{aligned} &\min_{U^s} \lambda_s \cdot J_s(U^s, x^s_{k_s}) \\ &\text{subject to} \\ &x^s(k_s+\ell+1) = f_s(x^s(k_s+\ell), u^s(k_s+\ell), X^s(k_s+\ell)), \\ &\qquad\qquad\qquad\qquad \ell = 0,1,\dots,N_s-1, \\ &X^s(k_s+\ell) = [x^1(k_s+\ell),\dots,x^{s-1}(k_s+\ell), \\ &\qquad u^1(k_s+\ell),\dots,u^{s-1}(k_s+\ell)], \quad \ell = 0,1,\dots,N_s-1, \\ &x^s(k_s) = x^s_{k_s}, \\ &x^s(k_s+\ell) \in \mathbf{X}^s, \quad \ell = 1,2,\dots,N_s, \\ &u^s(k_s+\ell) \in \mathbf{U}^s, \quad \ell = 0,1,\dots,N_s-1, \end{aligned} \tag{2.2}$$

where $U^s = [u^s(k_s)', \dots, u^s(k_s+N_s-1)']'$ is the sequence of future control actions in layer s, $J_s(U^s, x^s_{k_s}) = [J^s_1(U^s, x^s_{k_s}), \dots, J^s_{l_s}(U^s, x^s_{k_s})]'$ are the l_s objective functions to minimize, $\lambda_s = [\lambda_s^1, \dots, \lambda_s^{l_s}]$ is the weighting factor vector, N_s is the prediction horizon, $x^s(k_s+\ell)$ is the ℓ-step-ahead predicted state from the initial state $x^s_{k_s}$, $X^s(k_s+\ell)$ is a vector with the predicted states and the outputs of higher layers that will affect the dynamics of layer s at step $k_s+\ell$. We suppose $X^s(k_s+\ell)$ is known from the higher level MPC and is fixed for the optimization problem to solve in layer s. The state as well as the inputs are constrained to $\mathbf{X}^s$ and $\mathbf{U}^s$.

As can be noted, the optimizer at a higher level computes its desired control actions, which can be interpreted as reference signals for the immediately lower layers. The design of the higher level module plays a fundamental role. In fact, even if it is based on static optimization, the adopted model has to be periodically updated, while keeping coherence, by means of some adequate procedure to deal with changing operating conditions. To guarantee that these inputs are feasible, and to consider the presence of disturbances in the lower layers, the layers should communicate.

2.2.2 Hierarchical multiobjective model predictive control

Usually the objective functions in MPC are conflicting, i.e., a solution that optimizes one objective may not optimize others [22]. Multiobjective (MO) in MPC is a generalization of MPC, where instead of minimizing a single objective function, we consider more performance indices. In Hierarchical Multiobjective Model Predictive Control (HMO-MPC), if the layer s modeled by (2.1) has conflicts, the following multiobjective problem is solved:

$$\min_{U^s} \left\{ J_1^s(U^s, x_{k_s}^s), J_2^s(U^s, x_{k_s}^s), \ldots, J_{l_s}^s(U^s, x_{k_s}^s) \right\} \tag{2.3}$$

subject to the same constraints as in (2.2). The variables U^s and $J_l^s(U^s, x_{k_s}^s)$, $l = 1, \ldots, l_s$, are the sequence of future control actions and the objective functions to minimize in layer s respectively. The solution of HMO-MPC problem is a set of control action sequences called Pareto optimal set. Next we define Pareto optimality. Consider a feasible control sequence $U_P^s = [u_P^s(k_s)', \ldots, u_P^s(k_s + N_s - 1)']'$. The sequence U_P^s is said to be Pareto optimal if and only if there does not exist another feasible control action sequence U such that:

1) $J_i^s(U, x_k) \leq J_i^s(U_P^s, x_k)$, for $i = 1, \ldots, l_s$.
2) $J_j^s(U, x_k) < J_j^s(U_P^s, x_k)$, for at least one $j \in \{1, \ldots, l_s\}$.

The Pareto optimal set contains all Pareto optimal solutions. The set of all objective function values corresponding to the solutions is known as the Pareto optimal front. The relation between MPC and MO in MPC will be explained with a simple example. Let us consider a MPC problem that involves minimizing the single objective function $\lambda J_1(U, x_k) + (1 - \lambda) J_2(U, x_k)$, $\lambda \in (0, 1)$, and a MO-MPC problem that involves minimizing $\{J_1(U, x_k), J_2(U, x_k)\}$. As seen in Fig. 2.1, the MPC optimal solution U_{MPC}^* belongs to the Pareto solution set of the MO-MPC problem (more details of the conditions for this can be found in [22] and references within). If we solve the MPC problem for a wide range of weighting factor values λ, we would obtain an approximation of the same Pareto set for MO-MPC. The procedure should be repeated at every instant, which could become extremely inefficient in terms of computer resources. In this paper, we use explicit enumeration for the simulation results to measure the benefits of the approach. We claim that for bigger problems evolutionary multiobjective optimization algorithms could be used; however, it will be important to evaluate the effects of using a metaheuristic in the performance of the system, because the Pareto set is not always obtained.

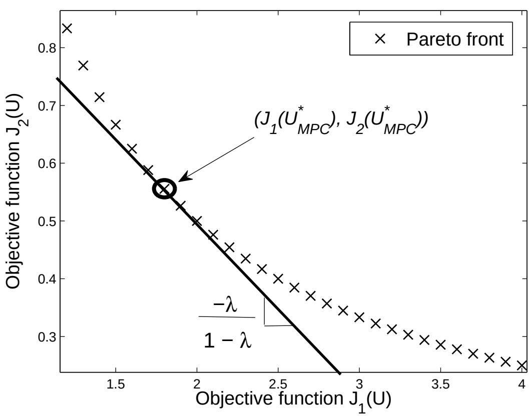


Figure 2.1: Relation between MPC and MO-MPC solutions

From the set of the optimal control solutions, just the first component $u^s(k_s)$ of one of those solutions has to be applied to the system, so at every instant, the controller (dispatcher in the context of a DPDP) has to use a criterion in order to find the control sequence that better suits the current objectives. In this paper, that decision is obtained after the Pareto set is determined. Then, it is not possible to choose a priori some weighting factor and to solve a single-objective optimization problem. The idea is to provide to the dispatcher a more transparent tool for the decisions.

2.3 Dynamic pickup and delivery problem

2.3.1 Process description

Dial-a-ride systems (DARS) are transit services which provide a shared-ride door-to-door service with flexible routes and schedules. The quality of service of a DARS is supposed to be in between of public transit buses and taxis. The typical specifications are the users pickup and delivery destinations and desired pickup or delivery times. We will assume that all the requests are known only after the dispatcher receives the associated call and that all the users want to be served as soon as possible. Thus, even we will not include explicitly hard windows, to provide a good service we propose a user-oriented objective function that deals with the problem of undesired assignments to clients, and keeps the service provided as regular (stable) as possible.

The service demand η_k comprises the information of the request and is characterized by two positions, pickup p_k and delivery d_k, the instant of the call t_k, a label r_k that identifies the passenger who is calling and the number of passengers waiting there Ω_k. Also we consider the expected minimum arrival time tr_k which is the best possible time to serve the passenger, considering a straight journey from origin to destination (like a taxi service) and a waiting time obtained with the closest available vehicle (in terms of capacity) to pick up that passenger.

We assume a fixed and known fleet size F over an urban area A. The dispatcher receives calls asking for service every instant k. Once a new request enters the system, the assignment of the vehicle and the insertion position of the new request into the previous sequence of that vehicle, are control actions decided by the dispatcher (controller), based on a dynamic objective function. Then, at any instant k, each vehicle j is assigned to complete a sequence of tasks which includes several points of pickup and delivery. Only one of those vehicles will serve the last new request. The set of sequences $u(k) = S(k) = \{S_1(k), ..., S_F(k)\}$ correspond to the control variable. The sequence of stops assigned to vehicle j at instant k is given by $S_j(k) = \left[S_j^0(k), S_j^1(k), ..., S_j^{w_j(k)}(k)\right]$, where $S_j^i(k)$ is the information about the i-th stop and $w_j(k)$ is the number of planned stops of vehicle j at instant k. The i-th stop information comprises the label of the user $r_j^i(k)$, the spatial coordinate $P_j^i(k)$, whether the stop is a pickup or delivery $z_j^i(k)$ and the number of users waiting at the i-th stop $\Omega_j^i(k)$.

Two sources of stochasticity are considered: the first regarding the unknown future demand entering the system in real-time, and the second coming from the network traffic conditions, in its spatial and temporal dimension represented by a speed distribution $v(t, p)$ at instant t in a position p. We will assume in this work a conceptual network, where the trajectories are defined as the straight line that joins two consecutive stops. Besides, a speed distribution for the urban zone during a typical period represented by a speed model $\hat{v}(t, p)$ is supposed to be known, which could be obtained from historical data.

2.3.2 Process model

The predictive model for vehicle j at instant k proposed in [12] is given by:

$$
\begin{aligned}
\hat{\chi}_j(k+1) &= \begin{cases} P_j^{i^*}(k) + \displaystyle\int_{t_k}^{t_k+\tau} \frac{\hat{v}(t,p(t))(P_j^{i^*+1}(k) - P_j^{i^*}(k))}{\|P_j^{i^*+1}(k) - P_j^{i^*}(k)\|} dt \\ \qquad\qquad\qquad \text{if} \quad i^* < w_j(k) \\ P_j^{i^*}(k) \qquad \text{if} \quad i^* = w_j(k) \end{cases} \\
\hat{T}_j^i(k+1) &= \begin{cases} T_j^0(k) \quad \text{if} \quad i = 0 \\ t_k + \displaystyle\sum_{s=1}^{i} \kappa_j^s(k) \quad \text{if} \quad i = 1,\dots,w_j(k) \end{cases} \\
\hat{L}_j^i(k+1) &= \begin{cases} L_j^0(k) \quad \text{if} \quad i = 0 \\ L_j^0(k) + \displaystyle\sum_{s=1}^{i} (2z_j^s(k) - 1)\Omega_j^s(k), i = 1,.,w_j(k) \end{cases}
\end{aligned}
\tag{2.4}
$$

where $\hat{\chi}_j(k)$ is the expected position of vehicle j, $\hat{T}_j^i(k)$ the expected departure time of vehicle j from stop i, and $\hat{L}_j^i(k)$ the expected load of vehicle j when leaving stop i. Moreover, t_k is the continuous instant time when request k happens, τ is the instant between t_k and the occurrence of the future probable call, i^* is the expected last stop visited by the vehicle before instant $t_k + \tau$, and $\kappa_j^i(k)$ is an estimation of the time interval between stop $i-1$ and stop i. Fig. 2.2 shows an example of a sequence assigned to vehicle j at instant k.

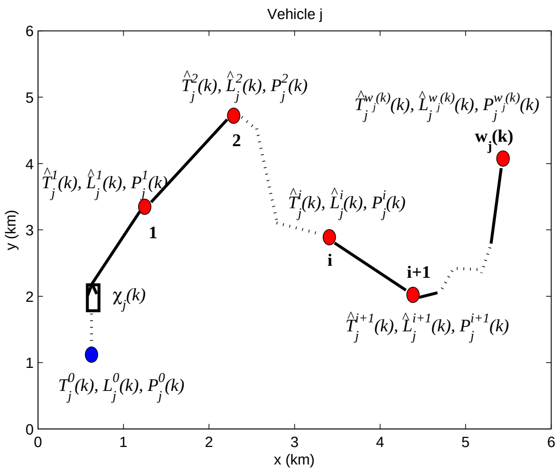


Figure 2.2: Representation of sequence of vehicle j and its stops

2.3.3 Objective functions

The motivation of this work is to provide to the dispatcher an efficient tool that captures the hierarchical structure of the DPDP problem and the trade-off between users and operator costs. Besides, we design an objective function able to reflect the fact that some users can become particularly annoyed if their service is postponed (either pickup or delivery), by means of an incremental weight in the objective function that penalizes differently very-long waiting or travel times.

The optimization variables are the current sequence $S(k)$ that incorporate the new request η_k, and the future sequences $S^h = \{S^h(k+1),\dots,S^h(k+N)\}$, $h = 1,\dots,h_{max}$, that incorporate the prediction of future requests. The scenario h consists of the sequential occurrence of $N-1$ estimated future request $\hat{\eta}^h_{k+1}, \hat{\eta}^h_{k+2}, \ldots, \hat{\eta}^h_{k+N-1}$, with a probability p_h. Thus $S_k^{k+N} = \{S(k), S^1, \ldots, S^{h_{\max}}\}$ comprises all the control actions to be calculated. The user cost $J_1(k)$ and the operator cost $J_2(k)$ are given by:

$$
\begin{aligned}
J_1(k) &= \sum_{j=1}^{F}\sum_{h=1}^{h_{max}}\sum_{\ell=1}^{N} p_h \cdot (J^U_{j,h}(k+\ell) - J^U_{j,h}(k+\ell-1)) \\
J^U_{j,h}(k+\ell) &= \\
\theta_v & \sum_{i=1}^{w_j(k+\ell)} f^v_{r^i_j(k+\ell)}(1 - z^i_j(k+\ell))(\hat{T}^i_j(k+\ell) - tr_{r^i_j(k+\ell)}) \\
&+\theta_e \sum_{i=1}^{w_j(k+\ell)} f^e_{r^i_j(k+\ell)} z^i_j(k+\ell)(\hat{T}^i_j(k+\ell) - t_{r^i_j(k+\ell)}),
\end{aligned}
\tag{2.5}
$$

$$
\begin{aligned}
J_2(k) &= \sum_{j=1}^{F}\sum_{h=1}^{h_{max}}\sum_{\ell=1}^{N} p_h \cdot (J^O_{j,h}(k+\ell) - J^O_{j,h}(k+\ell-1)) \\
J^O_{j,h}(k+\ell) &= c_T(\hat{T}^{w_j(k+\ell)}_j(k+\ell) - T^0_j(k+\ell)) \\
&\quad + c_L \sum_{i=1}^{w_j(k+\ell)} D^i_j(k+\ell),
\end{aligned}
\tag{2.6}
$$

where N is the prediction horizon, h_{max} is the number of predicted scenarios, $k+\ell$ is the instant at which the ℓ-th request enters the system, measured from instant k, p_h is the probability of occurrence of the h-th scenario, $J^U_{j,h}(\cdot)$ is the cost of the users in vehicle j, and $J^O_{j,h}(\cdot)$ is the operator cost of vehicle j when the scenario h occurs. The first component of $J^U_{j,h}(\cdot)$ is related to the re-routing time and the second component to the effective waiting time experienced by user $r^i_j(\cdot)$. Moreover, $f^v_{r^i_j(\cdot)}$ and $f^e_{r^i_j(\cdot)}$ are special weighting functions designed for the user $r^i_j(\cdot)$; both will start to grow linearly if the user is not experiencing a good total travel or a good waiting time respectively. Regarding $J^O_{j,h}(\cdot)$, it includes a first term that depends on the total operational time and another which depends on the total traveled distance. Thus, $D^i_j(\cdot)$ represents the distance between stops $i-1$ and i in the sequence of vehicle j. Finally, θ_v, θ_e, c_T and c_L are weights defined by the dispatcher.

As this optimization problem is big and NP-hard, we propose to exploit its inherent hierarchical structure, splitting the problem in smaller ones that work coordinated in different time scales. In the third layer of the proposed structure the conflicts between users and operator will be solved by the use of multiobjective model predictive control.

2.4 HMO-MPC for the DPDP

The DPDP is divided in three layers. In the first layer, variables like prices, economical factors, fleet size, etc., are determined based on economical criteria. The second layer characterizes each vehicle according to its coverage area and occupancy by providing parameters of membership functions of a fuzzy inference system [23]. In the third layer, whenever a request appears, the vehicles are routed by minimizing user and operator costs using MO-MPC. In Fig. 2.3 the proposed scheme for DPDP is shown and next each layer is explained. The structure of this diagram is like the proposed in [24].

First layer, Management

This layer represent a plant-wide optimization process. Outputs which are assumed constant in a period of about two or more hours are determined in this layer. Those outputs are some parameters

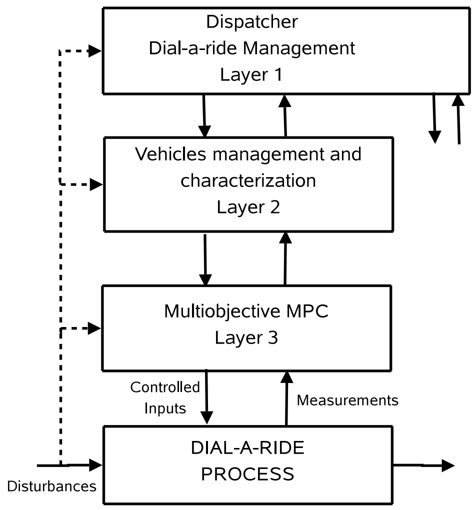


Figure 2.3: HMO-MPC for the DARS

of the objective functions (2.5) and (2.6) like the value of waiting θ_e and travel θ_v times for users, the value of each minute traveled by vehicles c_T, cost for kilometer traveled c_D, reasonable prediction horizon N, fleet size F, call-rate τ, etc. The demand patterns h and their probabilities are determined here by using fuzzy clustering as in [11], and the membership functions of the clusters are determined. Those parameters clearly change on time, each one with a different but slow rate. For example the cost per kilometer traveled c_D could change because of daily variations of the price of fuel, the demand patterns h if there is a special event in a stadium, etc. Thus an MPC problem like (2.2) with $s = 1$ could be solved, where the objective function should incorporate economical indexes. In this paper we are focusing on the operational process, so we will assume that the information provided by this layer is given and fixed. However, whether or not a static or dynamic optimization is good, the way each parameter is determined and the analysis of more complex situations are topics for further research.

Second layer, Vehicles characterization

This layer generates the information used to determine the group of vehicles with better chances to serve new requests. The information is updated every 20 minutes and will permit to reduce the computational effort when discarding vehicles too far away from new requests η_k or when their number of tasks is too high. The output of this layer are the parameters of three membership functions (MFs) for each vehicle j, which represent the coverage in axis x $\mu_x^j(\cdot)$, in axis y $\mu_y^j(\cdot)$ and the number of tasks $\mu_N^j(\cdot)$. The following MFs are used:

$$\begin{aligned} \mu_x^j(\eta_k^x) &= e^{-\frac{0.5(\eta_k^x - \bar{P}_j^x)^2}{(\sigma_j^x)^2}}, \quad \mu_y^j(\eta_k^y) = e^{-\frac{0.5(\eta_k^y - \bar{P}_j^y)^2}{(\sigma_j^y)^2}} \\ \mu_N^j(w_j(k)) &= \frac{1}{1 + e^{-(w_j(k) - c_j(t))}} \end{aligned} \tag{2.7}$$

where η_k^x and η_k^y are the x and y coordinates of the pickup or delivery of the new request η_k, $\bar{P}_j^x$ and $\bar{P}_j^y$ are the mean values and σ_j^x and σ_j^y are the standard deviations of coordinates x and y of the task assigned to vehicle j including the current position of the vehicle and the last stop visited. The variable $c_j(t)$ is the point of inflection of the sigmoidal membership function. The gaussian MF for the coordinates captures the fact that some vehicles will serve requests in specific zones with a small coverage area, and others with a wider coverage. Due to the fact that the parameters are uploaded, it is possible to change the kind of trip a vehicle is doing according to the new requirements. Regarding the sigmoidal (logit) shape of the MF for the number of tasks, the idea is to include the fact that when a vehicle is too saturated with future tasks, not only a bad service would be provided to the user, but also more computation time for solving the optimization problem would be required because it is NP-hard with the number of stops. We use $c_j(t) = 10$ for the simulation results, so vehicles with up to 10 tasks are still reasonable. The value of $c_j(t)$ could start to be reduced when the driver of vehicle j is going to get lunch, or by the end of his journey. Other characteristics like occupancy, number of annoyed passengers, worst served passenger total time, could be also used to rank the vehicles with more chances to serve new requests.

Third layer, Multiobjective optimization

The last layer consists of two components. The first one is a pre-processing algorithm where the optimization problem is reduced and conflicts between users and operator costs are detected. If there is a conflict, in the second component, we solve it by using MO-MPC.

The pre-processing algorithm is divided in three steps. In the first and second steps, by means of fuzzy inference systems, the most ad-hoc trip patterns $\bar{h}$ and fleet $\bar{F}$ are determined. In the third step we detect conflicts between users and operator by solving two MPC problems, to optimize just user cost and to optimize just operator cost. The algorithm is the following:

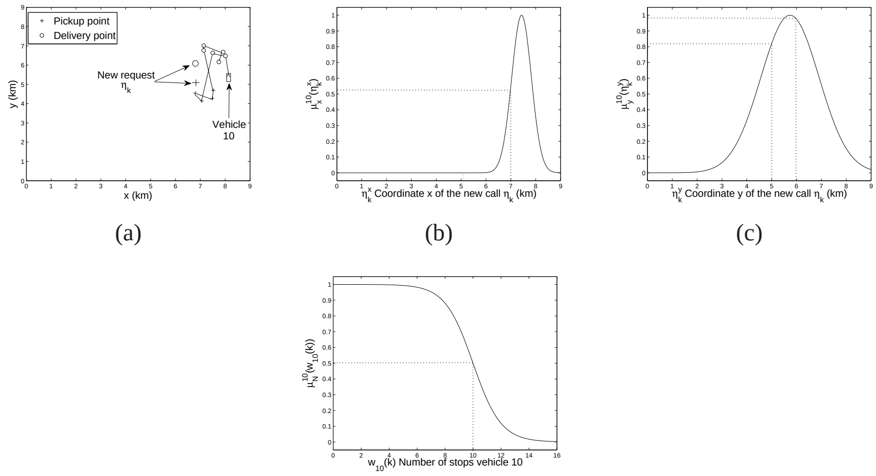
Step 1.1. Using a fuzzy inference system, the new request η_k is evaluated in the membership functions (MFs) of each trip pattern. This fuzzy inference uses the parameters of MFs provided by the first layer. The predicted future scenarios with a high activation degree are chosen. These are denoted by $\bar{h}$.

Step 1.2. With another fuzzy inference system, the candidate vehicles $\bar{F}$ to serve the new and the probable requests $\bar{h}$ are determined. To show how this fuzzy inference works, consider for example the vehicle 10 at instant k as shown in Fig. 4. This fuzzy inference uses the parameters of the MFs provided by the second layer. A new call η_k arrives, whose pickup coordinate is $(7,5)$ and whose delivery coordinate is $(7,6)$, as shown in Fig. 4(a). We check first whether η_k is in the coverage area of vehicle j, by evaluating the MFs $\mu_x^{10}(\cdot)$ and $\mu_y^{10}(\cdot)$ shown in Fig. 4(b) and Fig. 4(c) respectively. For the pickup we get $\mu_x^{10}(7)\cdot\mu_y^{10}(5) = 0.45$ and for the delivery $\mu_x^{10}(7)\cdot\mu_y^{10}(6) = 0.53$. Then we check whether the number of stops is big by using the MF $\mu_N^{10}(\cdot)$ shown in Fig. 4(d). At instant k, $w_j(k) = 10$, so $\mu_N^{10}(10) = 0.5$, which means that the vehicle is still having a reasonable number of tasks. Finally the activation degree of the rule for vehicle 10 equals 0.12. Whether vehicle 10 is a good candidate or not, will depend on the conditions of the other vehicles. The first vehicle candidate is obtained by choosing the vehicle with the maximum activation degree, the second candidate with the second maximum, and so on (defuzzification).

Step 1.3. Two MPC optimization problems are solved. To optimize just user cost and to optimize just operator cost:

$$\min_{S_k^{k+N}} J_1(k) = \sum_{j\in\bar{F}} \sum_{h\in\bar{h}} \sum_{\ell=1}^{N} p_h \cdot \Delta J_{j,h}^{U}(k+\ell) \qquad (2.8)$$
$$s.t. \quad \text{Model and constraints}$$

$$\min_{S_k^{k+N}} J_2(k) = \sum_{j\in\bar{F}} \sum_{h\in\bar{h}} \sum_{\ell=1}^{N} p_h \cdot \Delta J_{j,h}^{O}(k+\ell) \qquad (2.9)$$
$$s.t. \quad \text{Model and constraints}$$



(a) (b) (c)

(d)

Figure 2.4: Fuzzy characterization of vehicle 10. (a) Vehicle 10 sequence, Membership Functions for (b) x-axis, (c) y-axis and (d) number of stops

The objective functions minimized in (2.8) and (2.9), are like in (2.5) and (2.6) respectively, but considering just the set of vehicles $\overline{F}$ and the probable scenarios $\overline{h}$, which reduces the computational effort significantly. If the solution for both MPC problems (2.8) and (2.9) is the same or the trade-off between them is small, then the optimal solution which is closer to a pre-defined dispatcher criterion is used. If the trade-off is big, then MO-MPC is required to find the optimal Pareto front, as the set of vehicles has conflicts and a better picture of the trade-off is necessary.

The MO-MPC algorithm is divided in four steps. In the first step, for each vehicle $j \in \overline{F}$, the Pareto optimal sets for different conditions are determined. In the next step, the Pareto set for each scenario $h \in \overline{h}$ is obtained by coordinating different vehicles to serve all the requests. In the third step, the Pareto set for the MO-MPC problem is obtained. Finally in the last step a dispatcher selects a solution based on his/her criterion. Below each step is explained.

Step 2.1. The scenario h consists of the sequential occurrence of N requests $\eta_k, \hat{\eta}^h_{k+1}, \hat{\eta}^h_{k+2}, \ldots, \hat{\eta}^h_{k+N-1}$. For each vehicle $j \in \overline{F}$, for each scenario $h \in \overline{h}$, we will solve 2^N MO problems considering the cases where vehicle j is the one that serves none, one, or a combination of more of those requests. For example, if $N = 2$, for each vehicle we solve four MO problems considering the cases to serve none, to serve η_k, to serve $\hat{\eta}^h_{k+1}$, and to serve η_k and $\hat{\eta}^h_{k+1}$. The MO problem in this step is the following:

$$\min_{\{S_j(k), S^h_j(k+1), \ldots, S^h_j(k+N)\}} \left\{ \sum_{\ell=1}^{N} \Delta J^U_{j,h}(k+\ell), \sum_{\ell=1}^{N} \Delta J^O_{j,h}(k+\ell) \right\} \tag{2.10}$$

Capacity constraints and consistency are considered, so the Pareto set contains just feasible sequences. Note that some of those MO problems are easy to solve, but the more requests the vehicle serves, the more possible solutions we will have. In fact, considering the no-swapping constraint, the number of possible solutions when the N requests are served by vehicle j only is $0.5 \cdot \prod_{i=0}^{N-1}(w_j(k) + i)(w_j(k) + i - 1)$, where $w_j(k)$ is the number of stops of vehicle j and instant k. The MO problems in this step are the most time consuming, but they can be solved simultaneously and in parallel because they are not related with each other.

Step 2.2. Then for a given scenario $h \in \overline{h}$, considering the constraint that just one vehicle can serve each request, we obtain the Pareto set of the following MO problem:

$$\min_{\{S(k),S^1,\ldots,S^{h_{max}}\}} \left\{ \sum_{j\in\bar{F}} \sum_{\ell=1}^{N} \Delta J^{U}_{j,h}(k+\ell), \sum_{j\in\bar{F}} \sum_{\ell=1}^{N} \Delta J^{O}_{j,h}(k+\ell) \right\} \quad (2.11)$$

The solution of this MO problem is obtained with the Pareto sets from Step 2.1 by combining the $|\bar{F}|^N$ possible cases in a way that the current request and each future request are served by just one vehicle. For example, if we have three vehicles $\bar{F} = \{3,4,11\}$, for $N = 2$, the cases are $3-3$, $3-4$, $3-11$, $4-3$, $4-4$, $4-11$, $11-3$, $11-4$ and $11-11$, where $v_1 - v_2$ means that η_k is served by vehicle v_1 and $\hat{\eta}^h_{k+1}$ is served by vehicle v_2.

Step 2.3. Then, using the Pareto set of all the scenarios $h \in \bar{h}$, we solve the following MO problem:

$$\min_{S_k^{k+N}} \left\{ \sum_{j\in\bar{F}} \sum_{h\in\bar{h}} \sum_{\ell=1}^{N} p_h \cdot \Delta J^{U}_{j,h}(k+\ell), \sum_{j\in\bar{F}} \sum_{h\in\bar{h}} \sum_{\ell=1}^{N} p_h \cdot \Delta J^{O}_{j,h}(k+\ell) \right\} \quad (2.12)$$

The solution of this MO problem is obtained using the Pareto sets from Step 2.2, by multiplying each Pareto front by the probability of occurrence of the associated scenario p_h and then combining the different cases considering all the scenarios.

Step 2.4. The Pareto front from Step 2.3 is presented to the operator, who will select a sequence $S(k)$ that is Pareto optimal, based on a criterion. For example, the operator can choose the solution that provides the minimum user effective cost, or other characteristics that could be estimated. For estimating for example the effective user waiting time, we weight the expected waiting time of each of scenario with its probability of occurrence p_h. In this step the performance of the DPDP will depend on how good the criterion applied is.

In this kind of problems, HMO-MPC suits very well, as its main objective is to be implemented as a reference to support the decisions of the dispatcher, who has the flexibility of deciding which criterion is more adequate.

2.5 Simulation results

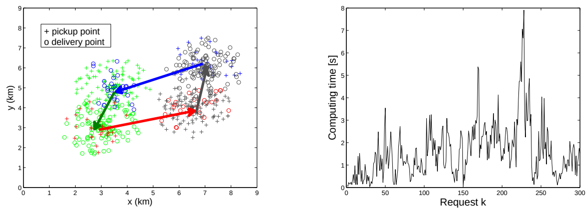
A period of four hours representative of a labor day (14:00-17:59) is simulated, over an urban area of approximately 81 (km^2). A fixed fleet of twenty vehicles is considered, with a capacity of four passengers each. We assume that the vehicles travel in a straight line between stops and that the transport network behaves according to a speed distribution with mean equal to $20(km/h)$. We suppose that the future calls are unknown for the dispatcher. However, he(she) has historical data from where the typical trip patterns can be extracted. A speed distribution model and the trip patterns are known, from the historical data and the fuzzy zoning method. This fuzzy zoning permits to generate the trip patterns and their probabilities as shown in Fig. 5(a) and Table I.

Three hundred calls were generated over the simulation period of four hours following the spatial and temporal distribution observed from the historical data. Regarding the temporal dimension, a negative exponential distribution for time intervals between calls with rate 0.8 $(call/min)$ was assumed. Regarding the spatial distribution, the pickup and delivery coordinates were generated randomly within each zone. The first 30 calls at the beginning and the last 30 calls at the end of the experiments were deleted from the statistics to avoid limit distortion (warm up period). One experiment was carried out to obtain the statistics, to show how the approach works. The experiment (emulating four hours and 300 on-line decisions) took $7.66(min)$, on average 1.69 seconds per request, on a Intel Core2 CPU, 3.00GHz processor. The computing time at each iteration is shown in Fig. 5(b). This

Table 2.1: Pickup and delivery coordinates and probabilities: 1st layer Fuzzy Zoning

$X_{pickup}(km)$	$Y_{pickup}(km)$	$X_{delivery}(km)$	$Y_{delivery}(km)$	Probability
3.0870	3.0244	6.5063	4.0556	0.1510
6.9598	5.8895	3.4377	4.9476	0.1510
3.4383	5.0403	3.0684	2.9579	0.3473
6.5473	3.9574	7.0399	5.9597	0.3506

computing time represents an upper bound of what is possible to do if a more efficient algorithm like the metaheuristics from the multiobjective evolutionary computation were applied (specially for the peaks, like the one of 8 seconds for request 248).



(a) (b)

Figure 2.5: HMO-MPC for the DPDP. (a) Origin-destination patterns, (b) Computation time

The objective function is formulated at two steps ahead, considering parameters $\theta_v = 16,7(\$/min)$, $\theta_e = 50(\$/min)$, $c_T = 25(\$/min)$, $c_L = 350(\$/Km)$. The users will start to get annoyed if their perceived total travel time is bigger than 1.7 times their minimal travel time, or if their waiting time is longer than $10(min)$. In the 3rd layer, the six best vehicles to serve a new request are chosen, ranked according the fuzzy inference system. The most likely demand pattern was used for the predictions.

The criteria for selecting a Pareto solution was the value nearest to a given user cost. We considered four cases: 500, 600, 700 and $800(Ch.\$)$ for cases a), b), c) and d) respectively.

Simulations for two steps ahead were conducted to analyze and evaluate the performance of the HMO-MPC strategies. In Table 2.2 the effective user waiting and travel time, user and operator costs are reported. In Table 2.3 we also show the number of passengers (pax) badly served, i.e., having a waiting time higher than a threshold of $10(min)$, as well as a very bad level of service considering the total time spent in the trip (in-vehicle and waiting time) and the worst served user cost. Tables 2.2 and 2.3 clearly show the trade off between opposite components. The resulting mean user cost over the whole simulation fitted quite well the thresholds defined at each case.

2.6 Conclusions and future work

A new approach to solve DPDP was presented. The proposed HMO-MPC scheme considers three layers. In the first one, variables with a long-term effect in the system are determined. In the second layer, the vehicles are characterized by fuzzy membership functions, which are used in the next layer to optimize in a better way the fleet. The last layer consists of a MO-MPC problem. Under the implemented on-line system it is easier and transparent for the operator to follow service policies as weighting parameters are not tuned.

Table 2.2: Simulation results, user and operator cost

Case	Effective travel time (min)	Effective waiting time (min)	User cost (Ch.$)	Operator cost (Ch.$)
a)	11.18	5.82	477.49	18124.56
b)	12.88	6.51	539.91	17499.78
c)	12.68	8.57	639.62	16910.75
d)	12.70	11.39	781.07	16670.66

Table 2.3: User indexes

Case	Waiting time higher than 10 min (pax)	Unfavorable total time (pax)	Worst served user cost (Ch.$)
a)	31	20	1679.82
b)	50	32	2000.56
c)	73	51	2574.48
d)	120	72	3256.77

In the other hand, the method we use in this paper has three main drawbacks [25]. First, to obtain the solution set from MO problem requires a big computational effort. Second, if the number of MO problems to solve is big, a lot of analysis and coordination will be required. Third, the adequacy and the knowledge of the decision-maker have a huge impact on the performance. For the first point, we claim that new toolboxes for Evolutionary Computation and other efficient algorithms like the proposed in [26] have been developed in recent years, so it is possible to determine a good representative pseudo-optimal Pareto set in a dynamic context. The second point is not even a problem in this paper, as we just have two opposite objectives, but in general, for more objectives further exploration and research are required. The last point is the same problem that also appears when properly tuning weights in a single objective function, i.e. having a good knowledge about the process is always important for obtaining a good control performance.

Future work will focus on efficient optimization algorithms. The coordination with buses, train, or other transport modes could also be a interesting topic.

Acknowledgment

This research has been supported by the European 7th framework STREP project "Hierarchical and distributed model predictive control (HD-MPC)", contract number INFSO-ICT-223854. Also by Fondecyt Chile Grant 1100239 and the Millennium Institute Complex Engineering Systems (ICM:P-05-004-F, CONICYT:FBO16).

Bibliography

[1] J. Desrosiers, F. Soumis and Y. Dumas, "A dynamic programming solution of a large-scale single-vehicle dial-a-ride with time windows", *American Journal of Mathematical and Management Sciences*, vol. 6, 1986, pp 301-325.

[2] M. Savelsbergh and M. Sol, "The general pickup and delivery problem", *Transportation Science*, vol. 29, 1995, pp 17-29.

[3] G. Berbeglia, J. Cordeau and G. Laporte, "Dynamic pickup and delivery problems", *European Journal of Operational Research*, vol. 202, 2009, pp 8-15.

[4] H. Psaraftis, "A dynamic programming solution to the single many-to-many immediate request dial-a-ride problem", *Transportation Science*, vol. 14, 1980, pp 130-154.

[5] H. Psaraftis, *Dynamic vehicle routing problems*, B.L. Golden and A.A. Assad editors, Vehicle routing methods and studies, 1988, pp 223-248.

[6] M. Gendreau, F. Guertin, J. Potvin and E. Taillard, "Parallel tabu search for real-time vehicle routing and dispatching", *Transportation Science*, vol. 33, 1999, pp 381-390.

[7] J. Cordeau and G. Laporte, "The dial-a-ride problem: Models and algorithms", *Annals of Operations Research*, vol. 153, 2007, pp 29-46.

[8] R. Dial, "Autonomous dial-a-ride transit: Introductory overview", *Transportation Research Part C: Emerging Technologies*, vol. 3, 1995, pp 261-275.

[9] S. Ichoua, M. Gendreau and J.Y. Potvin, "Exploiting knowledge about future demands for real-time vehicle dispatching", *Transportation Science*, vol. 40, 2006, pp 211-225.

[10] S. Mitrovic-Minic, R. Krishnamurti and G. Laporte, "Double-horizon based heuristic for the dynamic pickup and delivery problem with time windows", *Transportation Research Part B*, vol. 38, 2004, pp 669-685.

[11] D. Sáez, C.E. Cortés and A. Núñez, "Hybrid adaptive predictive control for the multi-vehicle dynamic pickup and delivery problem based on genetic algorithms and fuzzy clustering", *Computers & Operations Research*, vol. 35, 2008, pp 3412-3438.

[12] C.E. Cortés, D. Sáez and A. Núñez, "Hybrid adaptive predictive control for a dynamic pickup and delivery problem including traffic congestion", *International Journal of Adaptive Control and Signal*, vol. 22, 2008, pp 103-123.

[13] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - A review". *Journal of Process Control*, vol. 19, 2009, pp 723-731.

[14] L.D. Baskar, *Traffic management and control in intelligent vehicle highway systems*. PhD thesis, Delft University of Technology, Delft, The Netherlands, November 2009.

[15] A.N. Tarău, *Model-based control for postal automation and baggage handling*. PhD thesis, Delft University of Technology, Delft, The Netherlands, November 2010.

[16] J. Paquette, J. Cordeau and G. Laporte, "Quality of service in dial-a-ride operations", *Computers & Industrial Engineering*, vol. 56, 2009, pp 1721-1734.

[17] W.H. Yang, K. Mathur and R.H. Ballou, "Stochastic vehicle routing problem with restocking", *Transportation Science*, vol. 34, 2000, pp 99-112.

[18] K.C. Tan, C.Y. Cheong and C.K. Goh, "Solving multiobjective vehicle routing problem with demand via evolutionary computation", *European Journal of Operational Research*, vol. 177, 2007, pp 813-839.

[19] N. Jozefowiez, F. Semet and E. Talbi, "Multiobjective vehicle routing problems", *European Journal of Operational Research*, vol. 189, 2008, pp 293-309.

[20] D. Zambrano and E.F. Camacho, "Application of MPC with multiple objective for a solar refrigeration plant". *Proceedings of the 2002 International Conference on Control Applications*, vol. 2, 2002, pp 1230-1235.

[21] E.C. Kerrigan, A. Bemporad, D. Mignone, M. Morari and J.M. Maciejowski, "Multi-objective priorisation and reconfiguration for the control of constrained hybrid systems". *Proceedings of the American Control Conference*, Chicago Illinois, 2000, pp 1694-1698.

[22] A. Bemporad and D. Muñoz de la Peña, "Multiobjective model predictive control". *Automatica*, vol. 45, 2009, pp 2823-2830.

[23] Mamdani, E.H., "Applications of fuzzy logic to approximate reasoning using linguistic synthesis," IEEE Transactions on Computers, Vol. 26, No. 12, pp. 1182-1191, 1977.

[24] P. Tatjewski, "Advanced control and on-line process optimization in multilayer structures". *Annual Reviews in Control*, vol. 32, 2008, pp 71-85.

[25] Y. Haimes and D. Li, "Hierarchical multiobjective analysis for large-scale systems: Review and current status". *Automatica*, vol. 24, 1988, pp 53-69.

[26] R. Caballero, T. Gomez, M. Luque, F. Miguel and F. Ruiz, "Hierarchical generation of Pareto optimal solutions in large-scale multiobjective systems". *Computers & Operations Research*, vol. 29, 2002, pp 1537-1558.

Chapter 3

Game Theory Based Formulation of Distributed Model Predictive Control

The results in this chapter have been achieved by Felipe Valencia and Jairo Espinosa, Universidad Nacional de Colombia, Medellín, Colombia.

DMPC is a control scheme in which the system is divided into a number of subsystems. Each subsystem is able to share information with the other subsystems in order to determine its local control action [1, 2, 3, 4]. The main goal of the DMPC approach is to achieve some degree of coordination among agents that are solving local MPC problems with locally relevant variables, costs, and constraints, without solving the centralized MPC problem [5, 6, 7]. Compared with totally decentralized MPC schemes (noncentralized MPC controllers without information exchange), the global performance of the system is improved [5], [1], [2], but computational cost is increased due to communications, cooperation and maybe negotiation among subsystems [1].

Several approaches to the DMPC problem have been presented in the literature. In Table 3.1, advantages and disadvantages of the reviewed DMPC methods are summarized. In addition to the issues presented in Table 3.1, the approaches proposed in [5, 9, 17, 6, 10, 7, 12, 13] require the system to be stable and controllable. These requirements restrict the applicability of the proposed methods. In [14, 15, 16] the stability and controllability requirements are not considered, but the system should be stabilizable in order to apply the method proposed in these references.

Considering all these issues, game theory arises as an alternative to formulate and characterize the DMPC problem. Game theory is a branch of applied mathematics used in social sciences, economics, biology (particularly evolutionary biology and ecology), engineering, political science, international relations, computer science, and philosophy. Game theory attempts to capture behaviors in strategic situations, or games where the outcome of a player is function not only of his choices but also depends on the choices of others [18]. While initially developed to analyze competitions in which one individual does better at another's expense, it has been expanded to treat a wide class of interactions, which are classified according to several criteria. Today, "game theory is a sort of 'unified field' theory for the rational side of social science, where 'social' is interpreted broadly, to include human as well as non-human players (computers, animals, plants)"[19].

The first ideas of applying game theory to the DMPC problem are in [20, 21]. In these references the authors proposed a DMPC scheme based on Nash-optimality. In such approaches the DMPC problem was formulated as a non-cooperative game and it was demonstrated that the solution converged

References	Advantages	Disadvantages
[5, 8, 6]	Exchange of the predicted state trajectories to avoid communication problems.	The method is for independent subsystems linked only by the cost function.
[9, 7]	Reduction of the computational complexity of a DMPC problem.	The method requires that each local MPC problem can be solved only with local information.
[10, 11, 12, 13]	Reduction of the communication among subsystems.	Increment of the computational burden due to the solution of minimax local problems.
[14, 15, 16]	Each local cost function considers the effect of the local control inputs in the entire system behavior.	Subsystems are forced to cooperate, and the cooperation might steer the subsystems to operating points where they do not perceive any benefit

Table 3.1: Advantages and disadvantages of the reviewed DMPC methods

to the Nash equilibrium point of the game. However, in [14, 15, 16] the authors shown examples in which this approach produced an unstable closed-loop behavior.

After the works of Du and Li (see [20, 21]), Rantzer in [11, 12, 13] related the DMPC problem with the game theory by using the cooperative game approach presented by John von Neumann and Oskar Morgenstern in [22]. In the approaches presented by Rantzer, the Lagrange multipliers used in the dual decomposition methods (see [7] for details) were conceived as prices in a market mechanisms serving to achieve mutual agreements among subsystems. Based on previous conception of Lagrange multipliers, dynamic price mechanisms was used for decomposing and distributing the optimization associated with the original MPC problem. However, the approaches presented by Rantzer also converges to a Nash equilibrium point, with the same disadvantages presented in [14, 15, 16].

Other approaches related with the formulation of the DMPC problem as a game have been presented in [23, 24, 25]. In [23] a DMPC control scheme based on Nash optimality is also presented. In [24] an algorithm based on cooperative games for solving the DMPC problem was proposed. However, in this approach a real application of the concepts of game theory is not clear. Finally, in [25] the authors analyze the DMPC problem as a non-cooperative game. Here, properties like convergence and feasibility were derived based on the concept of Nash equilibrium point.

From the literature review, most approaches related with the application of game theory are based on non-cooperative games and on the application of the Nash optimality or Nash equilibrium point, with this related disadvantages in the control framework. In order to tackle this drawback (and the mentioned about the classic DMPC strategies), in this work we will assume that subsystems “bargain”among each other in order to (jointly) decide which strategy is the best with respect to their mutual benefit (as in [24]). The DMPC problem will be then reformulated as a n-person bargaining game based on the concepts presented by John Nash in [26, 27, 28] about such games. The n-person bargaining game involves n individuals that can collaborate for mutual benefit. The individuals communicate with each other in order to (jointly) decide which strategy is the best for each individual, based on the profit received under cooperative behavior [26]. So, in the proposed formulation, each subsystem is able to decide whether to cooperate or not with the other subsystems depending on the benefit received by the subsystem from the cooperative behavior. The selection of the bargaining approach was made because its main insight is focusing on others, i.e., to assess your added value, you have to ask not what other players can bring to you but what you can bring to other players [29].

This text is organized as follows: In Section 3.1 a mathematical background of symmetric bargaining games (based on the work of Nash [26, 27, 28]) is presented. Since the original bargaining game theory was formulated for static games an extension should be proposed. Such a extension and its application to the DMPC problem is also included in this section. Moreover, the properties of the DMPC formulated as a bargaining game are discussed. Based on [30], in Section 3.2 a mathematical background of nonsymmetric bargaining games is presented. An extension of such a theory to dynamic games also is proposed, and the properties of the DMPC formulated as a nonsymmetric bargaining game are discussed. In Section 3.3 the algorithm (or negotiation model) for implementing a distributed solution of both symmetric and nonsymmetric DMPC games is presented. The difference between the proposed algorithm and the other algorithms for DMPC (specifically the DMPC algorithms based on Lagrange multipliers) also are discussed. In Section 3.4 the conditions for the convergence and the stability of the proposed DMPC scheme are established. In Section 3.5 simulation results for both symmetric and nonsymmetric DMPC games are discussed. The symmetric case is illustrated using the quadruple tank system. The nonsymmetric case is illustrated using the hydro-power valley proposed in [31]. Finally, in Section 3.6 the concluding remarks are presented.

3.1 Distributed model predictive control as a symmetric bargaining game

Let us first introduce some notation used through the remainder of this paper. Let N be the set of players, $N = \{1,2,\dots,M\}$, $M \geq 2$. For $\alpha,\beta \in \mathbb{R}^M$, let $\alpha\beta$ denote the vector $[\alpha_1\beta_1,\dots,\alpha_M\beta_M]$, $\beta \geq \alpha$ denote the inequality $\beta_i \geq \alpha_i$ for every $i \in N$ (similarly for $\beta > \alpha$), and $\beta \leqslant \alpha$ denote $\beta_i \leqslant \alpha_i$ for every $i \in N$ (similarly for $\beta < \alpha$). For $T \subset \mathbb{R}^M$, let $\alpha T := \{\gamma \in \mathbb{R}^M \colon \gamma = \alpha\beta \text{ for some } \beta \in T\}$ and $\alpha + T = T + \alpha := \{\nu \in \mathbb{R}^M \colon \nu = \alpha + \beta \text{ for some } \beta \in T\}$. Also, for $a \in \mathbb{R}$, $a\alpha := [a\alpha_1,\dots,a\alpha_M]$ and $aT := \{a\alpha \colon \alpha \in T\}$.

A game is defined as the tuple $G = (N,\{\Omega_i\}_{i\in N},\{\phi_i\}_{i\in N})$, where $N = \{1,\dots,M\}$ is the set of players, Ω_i is a finite set of possible actions of player i, and $\phi_i : \Omega_1 \times \dots \times \Omega_M \longrightarrow \mathbb{R}$ is the payoff function of the ith player [32]. So, a DMPC problem can be defined as a tuple $G = (N,\{\Omega_i\}_{i\in N},\{\phi_i\}_{i\in N})$, where $N = \{1,\dots,M\}$ is the set of subsystems, Ω_i is the non-empty set of feasible control actions for subsystem i, and $\phi_i : \Omega_1 \times \dots \times \Omega_M \longrightarrow \mathbb{R}$, where ϕ_i is the cost function of the ith subsystem. Hence, a DMPC problem is a game in which the players are the subsystems, the actions are the control inputs, and the payoff of each subsystem is given by the value of its cost function.

Since it is assumed that the players are able to "bargain" in order to achieve a common goal, the game G can be analyzed as a bargaining game following the Nash theories about such games. A bargaining game is a situation involving a set of players who have the opportunity to collaborate for mutual benefit by an agreement on a joint plan of action [26, 28]. If an agreement is not possible, the players carry out an alternative plan which is determined by the information locally available. The benefit perceived by the player when an agreement is not possible is called disagreement point. Mathematically, a bargaining game is defined as follows [30]:

Definition 1 Bargaining Game:
A bargaining game for N is a pair (S,d) where:

1. *S is a nonempty closed subset of $\mathbb{R}^M$ (Closedness of the feasible set S is required for mathematical convenience.).*

2. $d \in \mathrm{int}(S)$, *d being the disagreement point.*

3. $\zeta_i(S) := \max\{\phi_i \colon (\phi_i)_{i\in N} \in S\}$ *exists for every* $i \in N$.

Here $\phi_i \colon \mathbb{R}^M \longrightarrow \mathbb{R}$ *denotes the profit function of player i for* $i = 1,\ldots,M$, *S denotes the feasible set of profit functions, and* $\zeta_i(S)$ *denotes the utopia point of subsystem i for* $i = 1,\ldots,M$. *Moreover, if the feasible set S is convex then the bargaining game* (S,d) *is called a convex bargaining game.*

Remark 1 *A bargaining game* (S,d) *is called symmetric if* $d_1 = d_2 = \ldots = d_M$, *and for every* $\phi \in S$ *any point* $\widetilde{\phi} \in \mathbb{R}^M$ *arising from* ϕ *by performing some permutation of its coordinates is also in S. If a bargaining game* (S,d) *does not satisfy these conditions, then it is called a nonsymmetric bargaining game [30].*

The outcome of a game (S,d) is a tuple $\varphi(S,d) = (\phi_1,\ldots,\phi_M)$ of profits received by the players. If any player does not cooperate then the corresponding position in $\varphi(S,d)$ is replaced by its disagreement point. Hence, if all subsystems decide not to cooperate $\varphi(S,d) = (d_1,\ldots,d_M)$. Thus, the problem is how do you get an outcome of the game (S,d) given that every player wants to maximize its own profit? With the purpose of solving this issue, Nash in [26, 28] stated that the solution $\varphi(S,d)$ should satisfy the following four axioms:

Axiom 1 *Symmetry:*
If (S,d) *is a symmetric bargaining game, then* $\varphi_1(S,d) = \ldots = \varphi_M(S,d)$.

Axiom 2 *Weak Pareto optimality:*
For $T \in \mathbb{R}^M$ *let*

$$W(T) := \{\alpha \in T \colon \text{ there is no } \beta \in T \text{ with } \beta > \alpha\}$$

denote the weakly Pareto optimal subset of T. *Then, for the game* (S,d), $\varphi(S,d) \in W(S)$.

Axiom 3 *Scale transformation covariance:*
For the game (S,d), *and all* $a,b \in \mathbb{R}^M$ *with* $a \geq 0$ *and* $(aS+b, ad+b)$, $\varphi(aS+b, ad+b) = a\varphi(S,d) + b$.

Axiom 4 *Independence of irrelevant alternatives:*
For all pair of games (S,d), (T,e), *if* $d = e$, $T \subset S$, *and* $\varphi(S,d) \in T$, *then* $\varphi(S,d) = \varphi(T,e)$.

Therefore, a function assigning to each player of the game (S,d) the maximum benefit, where the resulting tuple $\varphi(S,d)$ satisfies Axioms 1-4 is called Nash function of the game (S,d), and the tuple $\varphi(S,d)$ is called symmetric Nash bargaining solution of the game (S,d). From [33], the symmetric Nash bargaining solution of any game (S,d) is defined as follows:

Definition 2 *Symmetric Nash Bargaining Solution*
For every (S,d) *with convex feasible set, let* $\varphi(S,d)$ *be the outcome of* (S,d), *where the function*

$$\phi \longmapsto \Pi_{i\in N}(\phi_i - d_i)$$

is maximized over the set $\{\phi \in S \colon \phi \geq d\}$, *with* $\phi = (\phi_1,\ldots,\phi_M)$. *The solution* $\varphi(S,d)$ *is called the symmetric Nash bargaining solution of* (S,d), *and the product* $\Pi_{i\in N}(\phi_i - d_i)$ *is called the symmetric Nash product of the game* (S,d).

Remark 2 *Axioms 1-4 were proposed by Nash in order to characterize the outcome of two-person bargaining games. However, they have also been used to characterize the outcome of n-person bargaining games (see [32, 33, 30] and the references therein). Moreover, the solution presented in Definition 2 corresponds to an extension to n-person bargaining games of the solution proposed by Nash for two-person bargaining games.*

As a consequence of Axioms 1-4 and based on Definition 2, Proposition 1 and Lemma 1 arise.

Proposition 1 *The symmetric Nash bargaining solution $\varphi(S,d)$ of the bargaining game (S,d) is well defined, i.e., $\varphi(S,d)$ is unique, and it is possible to derive a geometrical characterization for $\varphi(S,d)$ from the symmetric Nash product $\Pi_{i\in N}(\phi_i - d_i)$.*

Proof 1 *See [30]*

Lemma 1 *Consider the bargaining game (S,d). Let $\gamma \in W(S)$ with $\gamma > d$. Then $\gamma = \varphi(S,d)$ if and only if*

$$\sum_{i\in N} \frac{\phi_i}{\phi_i - d_i} = \sum_{i\in N} \frac{\gamma_i}{\phi_i - d_i} \tag{3.1}$$

supports Θ at γ.

Proof 2 *See [30]*

Until here, the main elements of the axiomatic bargaining game theory proposed by Nash have been introduced. It is worth noting that the mathematical background presented in this section was developed for static games, i.e., the effect of the time is not considered in the decision of the players. Then, the concepts presented in this section should be redefined in a discrete-time dynamic context in order to analyze the DMPC problem as a bargaining game. In the following section the discrete-time dynamic bargaining game is defined, and the conditions of symmetry of such game are also introduced.

3.1.1 Symmetric discrete-time dynamic bargaining game

Since the axiomatic bargaining game theory has been developed in the static environment, few dynamic approaches of the original theory have been proposed in order to analyze dynamic bargaining games (see [34, 35, 36] and the references therein). However, these approaches focus mainly on developing procedures to find the coalition-formation-based solution of the game. Assuming that all controllers jointly decide which control action use at the same time, the coalition formation is only a consequence of the decision process and not an objective of the negotiation model of the controllers as in [32, 34, 35, 36]. In such a case (the DMPC problem), the axiomatic bargaining game theory brings an alternative to characterize the outcome of the game.

Let a discrete-time dynamic bargaining game refers to a situation where at each time step a static bargaining game (S,d) is solved depending on the dynamic evolution of the decision environment, where the dynamic evolution of the decision environment determined by a state vector $x(k) \in \mathbb{R}^n$ and by an input vector $u(k) \in \mathbb{R}^m$, with $x(k) \in \mathbb{X}$ and $u(k) \in \mathbb{U}$, $\mathbb{X}$ and $\mathbb{U}$ being the feasible sets for $x(k)$ and $u(k)$ respectively. In this game, we assume that the feasible set and/or the disagreement point can change with time. Mathematically, a discrete-time dynamic bargaining game is defined as follows:

Definition 3 *Discrete-time dynamic bargaining game:*
A discrete-time dynamic bargaining game for N *is a sequence of pairs* $\{(\Theta(0),\eta(0)),(\Theta(1),\eta(1)),\ldots\}$*, denoted by* $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ *(*$\eta(k)$ *being the disagreement point at time step* k*), where:*

1. $\Theta(k)$ *is a nonempty closed subset of* $\mathbb{R}^M$*, for* $k=1,2,3,\ldots$*.*

2. $\eta(k)\in\mathrm{int}(\Theta(k))$ *for* $k=1,2,3,\ldots$*,* $\eta(k)$ *being the disagreement point.*

3. $\zeta_i(\Theta(k)) := \max\{\phi_i(k)\colon (\phi_i(k))_{i\in N}\in\Theta(k)\}$ *exists for every* $i\in N$ *at each time step* k*, i.e., for* $k=1,2,3,\ldots$*.*

4. *There exists functions* $f_i\in\mathbb{R}^{n_i}, g_i\in\mathbb{R}, h_i\in\mathbb{R},\ i=1,\ldots,M$*, determining the dynamic evolution of the decision environment, the feasible set, and the disagreement point of player* i *such that*

$$\begin{aligned}x_i(k+1) &= f_i(x(k),u(k))\\ \Theta_i(k+1) &= g_i(x(k),u(k),\Theta(k))\\ \eta_i(k+1) &= h_i(x(k),u(k),\eta(k))\end{aligned}$$

   with $x_i(k)\in\mathbb{X}_i$*,* $\mathbb{X}_i\subset\mathbb{X}$*.*

5. *There exists a profit function* $\phi(x(k),u(k))\in\mathbb{R}^M$ *such that* $\phi(x(k),u(k))\in\Theta(k)$*.*

If g_i *is a convex function for* $i=1\ldots,M$*, then* $\Theta(k)$ *is convex and the game* $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ *is a convex discrete-time bargaining game.*

Remark 3 *From Definitions 1 and 3, and from Remark 1 a bargaining game* $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ *is called symmetric if* $\eta_1(k)=\ldots=\eta_M(k)$ *for* $k=0,1,2,\ldots,\infty$*, and for every* $\phi(k)\in\Theta(k)$ *any point* $\widetilde{\phi}(k)\in\mathbb{R}^M$ *arising from* $\phi(k)$ *by performing some permutation of its coordinates is also inside* $\Theta(k)$ *for* $k=0,1,2,\ldots$

Since a game $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ is a sequence of static bargaining games the outcome of such games is given by the sequence of solutions $\{\varphi(\Theta(0),\eta(0)),\varphi(\Theta(1),\eta(1)),\ldots\}$. Assume the game $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ symmetric and convex. Let $\{\varphi(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ denote the sequence of solutions of a game $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$, i.e., $\{\varphi(\Theta(k),\eta(k))\}_{k=0}^{\infty} := \{\varphi(\Theta(0),\eta(0)),\varphi(\Theta(1),\eta(1)),\ldots\}$ Let l denote a fixed time instance, i.e., $l=k$ for a fixed k. Then, based on Definition 2 $\{\varphi(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ is a sequence of elements of the set $\{(\phi_1(l),\ldots,\phi_M(l))\in\Theta(l)\colon \phi(l)\geq\eta(l))\},\ l=1,2,\ldots$, where the function

$$\phi(l)\longmapsto\Pi_{i\in N}(\phi_i(l)-\eta_i(l)) \tag{3.2}$$

is maximized at l. Note that the outcome of the game $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ at fixed time step l given by (3.2) has the same properties than the solution of the symmetric bargaining game (S,d), i.e., satisfies Axioms 1-4, Proposition 1, and Lemma 1. Moreover, under the assumption of symmetry of $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ these properties holds for $l=0,1,2,\ldots$ Below, the DMPC problem is formulated as a symmetric discrete-time dynamic bargaining game. Also, the properties satisfied by the DMPC formulated as a symmetric discrete-time dynamic bargaining game are discussed.

3.1.2 DMPC as a symmetric discrete-time dynamic bargaining game

Let us first introduce some notation used in this section. Let Ω_i denote the feasible set of control actions for subsystem i, $i = 1,\ldots,M$, defined as the Cartesian product $\Omega_i = \Pi_{j=0}^{N_u}\Lambda_i$, where Λ_i is the feasible set for the control actions $u_i(k+j)$, for $j = 0,\ldots,N_u$ determined by the physical and operational limits of subsystem i, with N_u being the control horizon. Let $\tilde{u}(k) = [\tilde{u}_1^T(k),\ldots,\tilde{u}_M^T(k)]^T$, where $\tilde{u}_i(k) = [u_i^T(k),\ldots,u_i^T(k+N_u)]^T$ for $i = 1,\ldots,M$. Assume that $0 \in \Lambda_i$ for $i = 1,\ldots,M$. Assume that Λ_i is closed, convex, and independent of k for $i = 1,\ldots,M$ (closedness of Λ_i is required for mathematical convenience). Note that $\Omega = \Pi_{i=1}^{M}\Omega_i$ is the feasible set for the whole system determined by the physical and operational constraints. Moreover, Ω is closed, convex, and independent of k.

Similar definition of the feasible sets Ω_i, Ω have been made in [9, 17, 7, 16, 14, 15]. Also, note that there have been formulated constraints over the inputs only. This was made under the assumption that the state constraints are time independent and can be expressed as input constraints using the prediction model. Systems like the quadruple tank system presented in [37], the two reactors chain followed by a flash separator presented in [38, 14] and the hydro-power valley presented in [31] are examples of systems satisfying the assumptions made to determine the sets Λ_i, Ω_i, Ω. For more examples of systems satisfying the assumptions about Λ_i, Ω_i, Ω see [16, 14, 15, 39].

Let $\phi_i(\tilde{u}(k);x(k))$ denote the cost function for subsystem i, $i = 1,\ldots,M$, where the notation $(\tilde{u}(k);x(k))$ indicates that the function ϕ_i depends on $\tilde{u}(k)$ and $x(k)$ is a parameter whose time evolution is given by the linear state update equation

$$x(k+1) = Ax(k) + Bu(k)$$

where A and B are obtained by linearizing the model describing the behavior of the whole system [14]. For the sake of simplicity of notation we will not indicate the dependence of ϕ_i on $x(k)$ explicitly in the remainder of this text and thus write $\phi_i(\tilde{u}(k))$ instead $\phi_i(\tilde{u}(k);x(k))$. Assume that $\phi_i(\tilde{u}(k))$ is a quadratic positive convex function of $\tilde{u}(k)$ for $i = 1,\ldots,M$ as in [37, 38]. Assume that the subsystems are able to "bargain" in order to achieve a common goal: to maintain both the local and the whole system performance by driving the states of the system to their reference values.

Let $\Upsilon(k) := \{\phi_i(\tilde{u}(k)) \colon \tilde{u}(k) \in \Omega,\ \forall i \in N\}$. Since Ω is time-invariant for $i = 1,\ldots,M$ the feasible set $\Upsilon(k)$ is also time invariant, i.e., $\Upsilon(1) = \Upsilon(2) = \ldots = \Upsilon$. Moreover, since Ω is closed and convex and by the continuity and convexity of $\phi_i(\tilde{u}(k))$ with respect to $\tilde{u}(k)$, the set Υ is closed and convex. Note that Υ defines a set of possible values of the cost function of every subsystem given the set Ω. So, it is only needed to define a disagreement point in Υ in order to complete the formulation of the DMPC problem as a symmetric discrete-time dynamic bargaining game.

From [28], the disagreement point should give to the players a strong incentive to increase their demands as much as possible without losing compatibility. Therefore, following this statement let us define the disagreement point $\eta(k) := (\eta_1(k),\ldots,\eta_M((k))$ as

$$\eta_i(k+1) = \begin{cases} \eta_i(k) - \alpha|\eta_i(k) - \phi_i(\tilde{u}(k))| \text{ if } \eta_i(k) > \phi_i(\tilde{u}(k)) \\ \eta_i(k) + \alpha|\phi_i(\tilde{u}(k)) - \eta_i(k)| \text{ if } \eta_i(k) < \phi_i(\tilde{u}(k)) \end{cases}$$

$\forall i \in N$, with $0 < \alpha < 1$. With this definition of the disagreement point, if subsystem i decides to cooperate then it can improve its expected performance by reducing the disagreement point in a factor $\alpha[\eta_i(k) - \phi_i(\tilde{u}(k))]$. But, if subsystem i decides not to cooperate its expected performance is increased by a factor $[\phi_i(\tilde{u}(k)) - \eta_i(k)]$ in order to make possible that subsystem i begins to cooperate after a few time steps.

Let the utopia point $\zeta_i(\Upsilon) := \min\{\phi_i(\tilde{u}(k)) \colon \phi_i(\tilde{u}(k)) \in \Upsilon\}$ exist for every $i \in N$. Then, the DMPC problem can be analyzed as a discrete-time dynamic bargaining game (denoted by $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$), with feasible set Υ, disagreement point $\eta(k)$, and utopia point $\zeta(\Upsilon)$. Note that in $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ only the disagreement point depends on the time step k, and that $\zeta_i(\Upsilon)$ is redefined because the objective of the DMPC is to minimize the cost function $\phi_i(\widetilde{u}(k))$, $\forall i \in N$. Moreover, based on Remark 3 a symmetric DMPC game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ can be defined as follows:

Definition 4 *Symmetric DMPC game:*
A DMPC game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ is symmetric if $\eta_1(l) = \eta_2(l) = \ldots = \eta_M(l)$ for $l = 0,1,2,\ldots$, and for every $\phi(\widetilde{u}(l)) \in \Upsilon$ any point $\widetilde{\phi}(\widetilde{u}(l)) \in \mathbb{R}^M$ arising from $\phi(\widetilde{u}(l))$ by performing some permutation of its coordinates is also in Υ.

Recall that $\phi_i(\widetilde{u}(k))$ is a quadratic function of $\widetilde{u}(k)$, then it can be written as [37, 38]

$$\phi_i(\widetilde{u}(k)) = \widetilde{u}^T(k)Q_{uui}\widetilde{u}(k) + x^T(k)Q_{xui}\widetilde{u}(k) + x^T(k)Q_{xxi}x(k)$$

Hence, if $\eta_1(k) = \ldots = \eta_M(k)$ for $k = 0,1,2,\ldots$, then a condition for which $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ is symmetric is that $Q_{uui} = Q_{uuj}$, $Q_{xui} = Q_{xuj}$, and $Q_{xxi} = Q_{xxj}$ for all $i,j \in N$. This condition comes from the following equality:

$$\widetilde{u}^T(k)Q_{uui}\widetilde{u}(k) + x^T(k)Q_{xui}\widetilde{u}(k) + x^T(k)Q_{xxi}x(k) = \widetilde{u}^T(k)Q_{uuj}\widetilde{u}(k) + x^T(k)Q_{xuj}\widetilde{u}(k) + x^T(k)Q_{xxj}x(k)$$

for all $i,j \in N$.

Now, assume the game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ symmetric. Since Υ is closed and convex, the objective of the DMPC is to minimize the cost function $\phi_i(\widetilde{u}(k))$ for all $i \in N$, and based on (3.2) the outcome of the game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ is a sequence of elements of the set $\{(\phi_1(\widetilde{u}(l)),\ldots,\phi_M(\widetilde{u}(l))) \in \Upsilon\colon \eta(l) \geq \phi(\widetilde{u}(l)), \widetilde{u}(l) \in \Omega,\ l = 1,2,\ldots\}$ where the function

$$\phi(\widetilde{u}(l)) \longmapsto \Pi_{i\in N}(\eta_i(l) - \phi_i(\widetilde{u}(l))) \tag{3.3}$$

is maximized at l.

Until here, the symmetric DMPC game and its outcome were defined. Now, we have to demonstrate that the solution of $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ given by (3.3) is the symmetric Nash bargaining solution of such a game. With this purpose, it is required to demonstrate that (3.3) satisfies Axioms 1-4 at a fixed time l, $l = 0,1,2,\ldots$ Let us begin with Proposition 2. This proposition is required for proving that the outcome of the game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ given by (3.3) is the symmetric Nash bargaining solution of such a game, and for establishing that the corresponding optimization problem is well-posed.

Proposition 2 *The solution $\varphi(\Upsilon,\eta(l))$ of the game $(\Upsilon,\eta(l))$ is unique at l if Υ and $\phi(\widetilde{u}(l))$ are both convex.*

Proof 3 *In order to prove Proposition 2, we have to demonstrate that the set*

$$L = \{\phi_i(\widetilde{u}(k)) \in \Upsilon\colon \eta(l) \geq \phi(\widetilde{u}(l)), \widetilde{u}(l) \in \Omega, i \in N\}$$

is convex, and that (3.3) is strictly quasiconcave on L. This guarantees that $\varphi(\Upsilon,\eta(l))$ is unique at l.

Note that $L = \Upsilon \cap O$, with $O = \{\phi_i(\widetilde{u}(l)) \in \mathbb{R}\colon \eta(l) > \phi(\widetilde{u}(l))\}$. Since Υ and O are both convex sets L is also a convex set. Then, the first part of the proof is completed.

For the second part of the proof, recall that a function $\vartheta\colon \Xi \longrightarrow \mathbb{R}$, with Ξ a convex subset of $\mathbb{R}^m$ for some $m \in \mathbb{N}$, is strictly quasiconcave if $\vartheta(\lambda\alpha + (1-\lambda)\alpha') > \min\{\vartheta(\alpha),\vartheta(\alpha')\}$ for any $\alpha \neq \alpha' \in \Xi$, $\lambda \in (0,1)$.

Let $\widetilde{u}(l), \widetilde{u}'(l) \in \Omega$ such that $\widetilde{u}(l) \neq \widetilde{u}'(l)$, $\phi(\widetilde{u}(l)) \in L$, and $\phi(\widetilde{u}'(l)) \in L$. Let $\vartheta(\widetilde{u}(l)) := \Pi_{i\in N}(\eta_i(l) - \phi_i(\widetilde{u}(l)))$. Then

$$\vartheta(\lambda\widetilde{u}(l) + (1-\lambda)\widetilde{u}'(l)) = \Pi_{i\in N}(\eta_i(l) - \phi_i(\lambda\widetilde{u}(l) + (1-\lambda)\widetilde{u}'(l)))$$

Since $\phi_i(\widetilde{u}(l))$ is a convex function of $\widetilde{u}(l)$

$$\phi_i(\lambda\widetilde{u}(l) + (1-\lambda)\widetilde{u}'(l)) < \lambda\phi_i(\widetilde{u}(l)) + (1-\lambda)\phi_i(\widetilde{u}'(l))$$

for $\lambda \in (0,1)$, $i = 1, \ldots, M$. Moreover,

$$\max\{\phi_i(\widetilde{u}(l)), \phi_i(\widetilde{u}'(l))\} > \lambda\phi_i(\widetilde{u}(l)) + (1-\lambda)\phi_i(\widetilde{u}'(l)) > \min\{\phi_i(\widetilde{u}(l)), \phi_i(\widetilde{u}'(l))\}$$

Hence,

$$\vartheta(\lambda\widetilde{u}(l) + (1-\lambda)\widetilde{u}'(l)) > \Pi_{i\in N}(\eta_i(l) - \max\{\phi_i(\widetilde{u}(l)), \phi_i(\widetilde{u}'(l))\}),$$

which implies

$$\vartheta(\lambda\widetilde{u}(l) + (1-\lambda)\widetilde{u}'(l)) > \min\{\vartheta(\widetilde{u}(l)), \vartheta(\widetilde{u}'(l))\}$$

Hence, the function $\vartheta(\widetilde{u}(l))$ is a strictly quasiconcave function of $\widetilde{u}(l)$ on L. Therefore, $\varphi(\Upsilon, \eta(l))$ is unique at l. □

Recall that the DMPC problem is originally a minimization problem. Then, let us redefine the weakly Pareto optimal subset of T as

$$W'(T) := \{\alpha \in T : \text{ there is no } \beta \in T \text{ with } \beta < \alpha\}$$

As a consequence of Proposition 2, Propositions 3 and 4 arise.

Proposition 3 *The bargaining solution $\varphi(\Upsilon, \eta(l))$ of the game $(\Upsilon, \eta(l))$ belongs to $W'(L)$ at l.*

Proof 4 *From Proposition 2 $\varphi(\Upsilon, \eta(l))$ is unique. Hence, does not exist $\widetilde{u}'(l) \in \Omega$ satisfying $\Pi_{i\in N}(\eta_i(l) - \phi_i(\widetilde{u}'(l))) > \Pi_{i\in N}(\eta_i(l) - \varphi_i(\Upsilon, \eta(l)))$ on L. Furthermore, since $\eta_i(l) - \phi_i(\widetilde{u}(l)) > 0$ on L does not exist $\widetilde{u}'(l) \in \Omega$ satisfying $\phi(\widetilde{u}'(l)) < \varphi(\Upsilon, \eta(l))$. So, $\varphi(\Upsilon, \eta(l)) \in W'(L)$ at l.* □

Proposition 4 *Let $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ and $\{(\Upsilon', \eta'(k))\}_{k=0}^{\infty}$ be a pair of DMPC games such that $\Upsilon' \subset \Upsilon$. Let $\eta'(l) = \eta(l)$, and $\varphi(\Upsilon, \eta(l)) \in \Upsilon'$ at l. Then $\varphi(\Upsilon, \eta(l)) = \varphi(\Upsilon', \eta'(l))$.*

Proof 5 *By Proposition 2 $\varphi(\Upsilon, \eta(l))$ is unique. Since $\Upsilon' \subset \Upsilon$, $\eta'(l) = \eta(l)$, and $\varphi(\Upsilon, \eta(l)) \in \Upsilon'$, the solution of the DMPC game $(\Upsilon', \eta'(l))$ at l is $\varphi(\Upsilon, \eta(l))$. Hence, $\varphi(\Upsilon, \eta(l)) = \varphi(\Upsilon', \eta'(l))$.* □

From Proposition 4 we can conclude that if $\eta(l) = \eta(l+1)$ and $x(l) = x(l+1)$, the solution $\varphi(\Upsilon, \eta(l))$ of $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ remains constant at $l+1$, i.e., $\varphi(\Upsilon, \eta(l)) = \varphi(\Upsilon, \eta(l+1))$. Moreover, following the procedure proposed in [30], and based on the Minkowski separation theorem [40, 41, 42], it is possible to derive a geometrical characterization for $\varphi(\Upsilon, \eta(l))$. Thus, by the quasiconcavity of (3.3) and by the geometrical characterization of $\varphi(\Upsilon, \eta(l))$ it is possible to establish that the outcome of the game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ is well-posed. Moreover, the geometrical characterization of $\varphi(\Upsilon, \eta(l))$ and Proposition 3 allow us to rewrite Lemma 1 as it is shown in Lemma 2 whose consequence is Proposition 5.

Lemma 2 *Consider a DMPC game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$. Let $\gamma(l)\in W'(L)$. Then $\gamma(l)=\varphi(\Upsilon,\eta(l))$ if and only if*

$$\sum_{i\in N}\frac{\phi_i(\widetilde{u}(l))}{\eta_i(l)-\phi_i(\widetilde{u}(l))}=\sum_{i\in N}\frac{\gamma_i(l)}{\eta_i(l)-\phi_i(\widetilde{u}(l))}$$

supports Υ at $\gamma(l)$ at l.

Proof 6 *See [30]*

Proposition 5 *Consider the DMPC game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$. For all $a,\ b\in\mathbb{R}^M$ with $a\geq 0$, the bargaining solution of the game $(a\Upsilon+b,a\eta(l)+b)$ at l is equal to $a\varphi(\Upsilon,\eta(l))+b$, i.e., $\varphi(a\Upsilon+b,a\eta(l)+b)=a\varphi(\Upsilon,\eta(l))+b$.*

Proof 7 *By definition $a\Upsilon+b=\{\widetilde{\phi}(\widetilde{u}(l))\in\mathbb{R}^M:\ \widetilde{\phi}(\widetilde{u}(l))=a\phi(\widetilde{u}(l))+b,\ \phi(\widetilde{u}(l))\in\Upsilon\}$. From Lemma 2, $\gamma(l)$ is the bargaining solution of the DMPC game $(\Upsilon,\eta(l))$ at l if and only if*

$$\sum_{i\in N}\frac{\phi_i(\widetilde{u}(l))}{\eta_i(l)-\phi_i(\widetilde{u}(l))}=\sum_{i\in N}\frac{\gamma_i(l)}{\eta_i(l)-\phi_i(\widetilde{u}(l))}$$

supports Υ at $\gamma(l)$. For the game $(a\Upsilon+b,a\eta(l)+b)$ we have

$$\sum_{i\in N}\frac{\widetilde{\phi}_i(\widetilde{u}(l))}{a_i\eta_i(l)+b_i-\widetilde{\phi}_i(\widetilde{u}(l))}=\sum_{i\in N}\frac{\phi_i(\widetilde{u}(l))}{\eta_i(l)-\phi_i(\widetilde{u}(l))}+\sum_{i\in N}\frac{b_i}{a_i(\eta_i(l)-\phi_i(\widetilde{u}(l)))},$$

which can be written equivalently as

$$\sum_{i\in N}\frac{\widetilde{\phi}_i(\widetilde{u}(l))}{a_i\eta_i(l)+b_i-\widetilde{\phi}_i(\widetilde{u}(l))}=\sum_{i\in N}\frac{\gamma_i(l)}{\eta_i(l)-\phi_i(\widetilde{u}(l))}+\sum_{i\in N}\frac{b_i}{a_i(\eta_i(l)-\phi_i(\widetilde{u}(l))},$$

and consequently

$$\sum_{i\in N}\frac{\widetilde{\phi}_i(\widetilde{u}(l))}{a_i\eta_i(l)+b_i-\widetilde{\phi}_i(\widetilde{u}(l))}=\sum_{i\in N}\frac{a_i\gamma_i(l)+b_i}{a_i\eta_i(l)+b_i-\widetilde{\phi}_i(\widetilde{u}(l))}$$

Then (by Lemma 2), the bargaining solution of the DMPC game $(a\Upsilon+b,a\eta(l)+b)$ at l is equal to $a\varphi(\Upsilon,\eta(l))+b$, i.e., $\varphi(a\Upsilon+b,a\eta(l)+b)=a\varphi(\Upsilon,\eta(l))+b$. □

Finally, by the symmetry of $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ we have:

Proposition 6 *A symmetric DMPC game $(\Upsilon,\eta(l))$ satisfies $\varphi_1(\Upsilon,\eta(l))=\varphi_2(\Upsilon,\eta(l))=\ldots=\varphi_M(\Upsilon,\eta(l))$ at l.*

Proof 8 *Following the procedure proposed in [30] for this demonstration we have: By the symmetry of the DMPC game $(\Upsilon,\eta(l))$ we have $\eta_1(l)=\ldots=\eta_M(l)$. Let $\widetilde{\varphi}(\Upsilon,\eta(l))$ denote the solution of the game $(\Upsilon,\eta(l))$ arising by the permutation of the elements of $\varphi(\Upsilon,\eta(l))$. By the definition of $W'(\Upsilon)$, $\varphi(\Upsilon,\eta(l))=\widetilde{\varphi}(\Upsilon,\eta(l))$. Then, $\varphi_1(\Upsilon,\eta(l))=\varphi_2(\Upsilon,\eta(l))=\ldots=\varphi_M(\Upsilon,\eta(l))$.* □

Propositions 3-6 imply the following theorem:

Theorem 1 *At l, the bargaining solution $\varphi(\Upsilon,\eta(l))$ of the DMPC game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ is the Nash bargaining solution of such a game.*

Proof 9 *From Propositions 3-6 the bargaining solution $\varphi(\Upsilon, \eta(l))$ of the DMPC game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ satisfies Axioms 1-4 at l. Then $\varphi(\Upsilon, \eta(l))$ is the Nash bargaining solution of the DMPC game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ for such time step.* $\square$

Theorem 1, Proposition 2, Lemma 2, and the geometrical characterization of $\varphi(\Upsilon, \eta(l))$ summarize the properties of the outcome of the game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$. Note that this solution can be computed as a solution of the maximization problem

$$\begin{aligned} & \max_{\widetilde{u}(l)} \Pi_{i\in N}(\eta_i(l) - \phi_i(\widetilde{u}(l))) \\ \text{subject to: } & \eta(l) > \phi(\widetilde{u}(l)) \\ & \widetilde{u}(l) \in \Omega \end{aligned} \tag{3.4}$$

which can be written equivalently as (3.5).

$$\begin{aligned} & \max_{\widetilde{u}(l)} \sum_{i=1}^{M} \log(\eta_i(l) - \phi_i(\widetilde{u}(l))) \\ \text{subject to: } & \eta(l) > \phi(\widetilde{u}(l)) \\ & \widetilde{u}(l) \in \Omega \end{aligned} \tag{3.5}$$

Let $\sigma_i(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)) = \phi_i(\widetilde{u}(k))$ for $i = 1, \ldots, M$, where $\widetilde{u}_{-i}(k) = [\widetilde{u}_1^T(k), \ldots, \widetilde{u}_{i-1}^T(k), \widetilde{u}_{i+1}^T(k), \ldots, \widetilde{u}_M^T(k)]$. Then, maximization problem (3.5) can be solved in a distributed way by locally solving the systemwide control problem (3.6).

$$\begin{aligned} & \max_{\widetilde{u}_i(k)} \sum_{r=1}^{M} \log(\eta_r(k) - \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))) \\ & \text{Subject to:} \\ & \eta_r(k) > \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)) \\ & \widetilde{u}_i(k) \in \Omega_i \end{aligned} \tag{3.6}$$

Maximization problem (3.6) is equivalent to maximization problem (3.5), considering fixed $\widetilde{u}_{-i}(k)$ and optimizing only in the direction of $\widetilde{u}_i(k)$. This formulation allows to each subsystem take into account the effect of its decisions in the behavior of the whole system and to promote the cooperation among subsystems. The convexity and feasibility of optimization problem (3.6) are analyzed in [38]. In Section 3.3 the algorithm for implementing (3.6) is presented. In the next Section the DMPC problem is analyzed as a nonsymmetric bargaining game. This is a more general case of DMPC problems than the symmetric case presented in this section (the symmetry conditions in Remark 3 are not considered).

3.2 Distributed model predictive control as a nonsymmetric bargaining game

As it is shown in Section 3.1, in order to derive a solution for a bargaining game an axiomatic approach was proposed in [28]. Such a characterization is based on the symmetry of the bargaining game. Recall that a bargaining game (S, d) is called symmetric if $d_1 = d_2 = \ldots = d_M$, and for every $\phi \in S$ any point $\widetilde{\phi} \in \mathbb{R}^M$ arising from ϕ by performing some permutation of its coordinates is also in S. If a bargaining game (S, d) does not satisfy these conditions, then it is called a nonsymmetric bargaining game.

Consequently with the definition of nonsymmetric bargaining game, Axiom 1 is not satisfied for such games. Then, additional axioms should be included in order to characterize the outcome of nonsymmetric games. In such a way, Axioms 5 and 6 have been proposed in [30] and [33] respectively (Axiom 6 is an example of the general principle of the game theory [43, 30]).

Axiom 5 *Individual Rationality:*
For every bargaining game (S,d) *the outcome of the game* $\varphi(S,d)$ *satisfies* $\varphi(S,d) > d$.

Axiom 6 *Consistency:*
For $\alpha \in \mathbb{R}^M$ *and* $\varnothing \neq L \subset N$, α_L *denotes the vector in* $\mathbb{R}^L$ *obtained by deleting the coordinates of* α *belonging to* $N \backslash L$. *For* $T \in \mathbb{R}^M$, $T_L := \{\beta_L \colon \beta \in T\}$, $(T,\alpha)_L := \{\beta_L \colon \beta \in T,\ \beta_{N\backslash L} = \alpha_{N\backslash L}\}$. *For a game* (T,d) *and a solution* φ, $(T,d,\varphi)_L := (T,\varphi(T,d))_L$, *denote the utility* $|L|$*-tuple available for the collective* L, *if the subsystem* i *outside* L *receive* $\varphi_i(T,d)$. *Then, for all games* $(S,d),(T,d)$, *and every* $\varnothing \neq L \subset N$ *if* $(S,d,\varphi)_L = (T,d,\varphi)_L,\ \varphi_L(S,d) = \varphi_L(T,d)$.

Let $\mathbb{R}^M_{++} := \{\alpha \in \mathbb{R}^M \colon \alpha_i > 0, \text{ for all } i \in N\}$. Let H denote a wighted hierarchy of N, i.e., H is an ordered $(m+1)$-tuple $H = \langle N^1,\ldots,N^m,w\rangle$, where $(N^1,\ldots,N^m)$ is a partition of N (i.e., the sets N^j, $j = 1,\ldots,m$ are pairwise disjoint nonempty sets whose union equals to N), and $w \in \mathbb{R}^M_{++}$ with $\sum_{i\in N^j} w_i = 1$ for every $j = 1,\ldots,m$ [30]. Let $P(T) := \{\alpha \in T \colon \text{ there is no } \beta \in T \text{ with } \beta \geq \alpha,\ \beta \neq \alpha\}$ denotes the Pareto optimal subset of T. Let $L_+(T,\gamma) := \{i \in L \colon \text{ there exists } \alpha \in T \text{ with } \alpha_i > \gamma_i\}$. Let $\arg\max\{f(\alpha) \colon \alpha \in T\} := \{\alpha \in T \colon f(\alpha) \geq f(\beta) \text{ for all } \beta \in T\}$.

Then, taking in mind Axioms 2-6 and the definition of weighted hierarchy a bargaining solution for a nonsymmetric game (S,d) is associated by lexicographically maximizing "Nash products" according to the partitions and the weights in H [30].

Definition 5 *Nonsymmetric bargaining solution [30, Definition 2.14]:*
Let $H = \langle N^1,\ldots,N^m,w\rangle$ *be a weighted hierarchy of* N. *Let* S^j, $j = 0,\ldots,m$ *denote the feasible set for the partition* N^j. *Then, the sets* S^j *are defined as follows:*

$$S^0 := \{\phi \in \mathbb{R}^M \colon \phi \in P(S), \phi \geq d\}$$

$$S^1 := \arg\max\{\Pi(\phi_i - d_i)^{w_i} \colon i \in N^1,\ \phi \in S^0\}$$

$$S^2 := \begin{cases} \arg\max\{\Pi(\phi_i - d_i)^{w_i} \colon i \in N^2_+(S^1,d),\ \phi \in S^1\} & \textit{if } N^2_+(S^1,d) \neq \varnothing \\ S^1 & \textit{otherwise} \end{cases}$$

$$\vdots$$

$$S^j := \begin{cases} \arg\max\{\Pi(\phi_i - d_i)^{w_i} \colon i \in N^j_+(S^{j-1},d),\ \phi \in S^{j-1}\} & \textit{if } N^j_+(S^{j-1},d) \neq \varnothing \\ S^{j-1} & \textit{otherwise} \end{cases}$$

$$\vdots$$

$$S^m := \begin{cases} \arg\max\{\Pi(\phi_i - d_i)^{w_i} \colon i \in N^m_+(S^{m-1},d),\ \phi \in S^{m-1}\} & \textit{if } N^m_+(S^{m-1},d) \neq \varnothing \\ S^{m-1} & \textit{otherwise} \end{cases}$$

Remark 4 *The solution* $\varphi^H(S,d)$ *of the lexicographic maximization problem of Definition 5 over the set* $\{(\phi_i \in S \colon d \leqslant \phi,\ i \in N\}$ *is the solution of the game* (S,d) *corresponding to the weighted hierarchy* H *and the product* $\Pi(\phi_i - d_i)^{w_i}$ *is called the nonsymmetric Nash product of the game* (S,d) *corresponding to the weighted hierarchy* H.

From Definition 5 Lemma 3 arises. This Lemma implies that the solution of a nonsymmetric game (S,d) assigning the unique element of S^m is well-defined.

Lemma 3 *Let H be a weighted hierarchy associated with a nonsymmetric game (S,d). Let S^m be the set defined in Definition 5. Then $|S^m| = 1$, i.e., S^m only has one element.*

Proof 10 *See [30]*

In addition to Lemma 3 it is possible to demonstrate that $\varphi^H(S,d)$ satisfies Axioms 2-6, hence is the Nash bargaining solution of a nonsymmetric bargaining game (S,d) [30]. Also, $\varphi^H(S,d)$ satisfies Proposition 1 and Lemma 1.

Similar than in the symmetric case, the axiomatic game theory developed for nonsymmetric games have been developed for static games. Then, in order to analyze a DMPC problem as a nonsymmetric bargaining game the concept of nonsymmetric discrete-time dynamic bargaining game should be introduced. Furthermore, some concepts presented until here in this section should be redefined because the DMPC problem is focused on the minimization of the cost function associated with each subsystem.

3.2.1 DMPC as a nonsymmetric discrete-time dynamic bargaining game

In Remark 3 conditions for the symmetry of discrete-time dynamic bargaining games were presented. These conditions establish that for discrete-time dynamic bargaining games, if $\eta_1(k) = \ldots = \eta_M(k)$ for $k = 0,1,2,\ldots$, and for every $\phi(k) \in \Theta(k)$, any point $\widetilde{\phi}(k) \in \mathbb{R}^M$ arising from $\phi(k)$ by performing some permutation of its coordinates is also inside $\Theta(k)$ for $k = 0,1,2,\ldots$, the game $\{(\Theta(k),\eta(k))\}_{k=0}^{\infty}$ is called symmetric. Such conditions are satisfied when $f_i(x(k),u(k)) = f_j(x(k),u(k))$, $g_i(x(k),u(k)) = g_j(x(k),u(k))$, $h_i(x(k),u(k)) = h_j(x(k),u(k))$, and $\mathbb{X}_i = \mathbb{X}_j$ for all $i, j \in N$. However, for real DMPC applications the symmetry conditions discrete-time dynamic bargaining games are heavily restrictive, mainly because real large-scale systems are composed by several different subsystems with different time evolution equations. Then, in general a DMPC game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ is nonsymmetric.

Let us redefine the Pareto optimal subset of T as $P(T) := \{\alpha \in T:$ there is no $\beta \in T$ with $\beta \leqslant \alpha,\ \beta \neq \alpha\}$. Also let us redefine the set $L_+(T,\gamma)$ as $L_+(T,\gamma) := \{i \in L:$ there exists $\alpha \in T$ with $\alpha_i < \gamma_i\}$. Moreover, Axiom 5 should be rewritten as follows:

Axiom 7 *DMPC Individual Rationality:*
For every bargaining game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ the outcome of the game $\{\varphi(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ satisfies the inequality $\varphi(\Upsilon,\eta(k)) < \eta(k)$ for $k = 1,2,\ldots$.

Based on these modifications to the original theory, the outcome of a game $\{(\Upsilon,\eta(k))\}_{k=0}^{\infty}$ with weighted hierarchy H is a sequence $\{\varphi^H(\Upsilon,\eta(0)),\varphi^H(\Upsilon,\eta(1)),\ldots\}$ denoted by $\{\varphi^H(\Upsilon,\eta(k))\}_{k=0}^{\infty}$, where for a fixed l, $\varphi^H(\Upsilon,\eta(l))$ is given by the solution of the lexicographic optimization problem

$$
\begin{aligned}
&\Upsilon^0 := \{\phi(\widetilde{u}(l)) \in \mathbb{R}^M : \phi(\widetilde{u}(l)) \in P(\Upsilon), \phi(\widetilde{u}(l)) \leqslant \eta(l)\}\\
&\Upsilon^1 := \arg\max\{\Pi(\eta_i(l) - \phi_i(\widetilde{u}(l)))^{w_i} :\ i \in N^1,\ \phi(\widetilde{u}(l)) \in \Upsilon^0\}\\
&\Upsilon^2 := \begin{cases} \arg\max\{\Pi(\eta_i(l) - \phi_i(\widetilde{u}(l)))^{w_i} :\ i \in N_+^2(\Upsilon^1,\eta(l)),\ \phi(\widetilde{u}(l)) \in \Upsilon^1\} \text{ if } N_+^2(\Upsilon^1,\eta(l)) \neq \varnothing \\ \Theta^1 \quad \text{otherwise} \end{cases}\\
&\vdots\\
&\Upsilon^m := \begin{cases} \arg\max\{\Pi(\eta_i(l) - \phi_i(\widetilde{u}(l)))^{w_i} :\ i \in N_+^m(\Upsilon^{m-1},\eta(l)),\ \phi(\widetilde{u}(l)) \in \Upsilon^{m-1}\} \text{ if } N_+^m(\Upsilon^{m-1},\eta(l)) \neq \varnothing \\ \Upsilon^{m-1} \quad \text{otherwise} \end{cases}
\end{aligned}
\tag{3.7}
$$

Remark 5 *Although the definition of weighted hierarchy requires the selection of the weights for each subsystem, there are not guidelines for choosing their values. In the control theory field, the values of the weights can be arbitrarily selected as* $w_i = \frac{1}{M},\ i = 1,\ldots,M$ *(such a selection is made in [9, 44, 16, 15]). However, performing controllability and/or sensitivity analysis can help to derive guidelines for the selection of the weights* w_i.

Let $H = \langle N, w\rangle$. Then, in view of the lexicographic solution (3.7) the nonsymmetric bargaining solution of a DMPC game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$ at time step k can be computed in a centralized way as a solution of the maximization problem (3.8). This solution has the same properties as the solution given by (3.4), except for Axiom 1.

$$\begin{aligned} &\max_{\widetilde{u}(k)} \Pi_{i=1}^{M}(\eta_i(k) - \phi_i(\widetilde{u}(k)))^{w_i} \\ \text{Subject to:}& \\ &\eta_i(k) > \phi_i(\widetilde{u}(k)) \\ &\widetilde{u}(k) \in \Omega \end{aligned} \tag{3.8}$$

Maximization problem (3.8) can be written equivalently as (3.9).

$$\begin{aligned} &\max_{\widetilde{u}(k)} \sum_{i=1}^{M} w_i \log(\eta_i(k) - \phi_i(\widetilde{u}(k))) \\ \text{Subject to:}& \\ &\eta_i(k) > \phi_i(\widetilde{u}(k)) \\ &\widetilde{u}(k) \in \Omega \end{aligned} \tag{3.9}$$

Then, maximization problem (3.8) can be solved in a distributed way by locally solving the systemwide control problem (3.10).

$$\begin{aligned} &\max_{\widetilde{u}_i(k)} \sum_{r=1}^{M} w_r \log(\eta_r(k) - \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))) \\ \text{Subject to:}& \\ &\eta_r(k) > \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)) \\ &\widetilde{u}_i(k) \in \Omega_i \end{aligned} \tag{3.10}$$

Note that maximization problem (3.10) is equivalent to maximization problem (3.9), considering fixed $\widetilde{u}_{-i}(k)$ and optimizing only in the direction of $\widetilde{u}_i(k)$. This formulation allows each subsystem to take into account the effect of its decisions in the behavior of the whole system and to promote the cooperation among subsystems. The convexity and feasibility of (3.10) is presented in [38]. In the next Section, the algorithm for implementing a distributed solution of DMPC games presented in Section 3.1 and Section 3.2 is presented.

3.3 Negotiation model

A negotiation model is a sequence of steps for computing the outcome of a game. In the literature several negotiation models have been proposed for solving n-person games, almost all of them based on the coalition formation (see [32] and the references therein for more complete information). The negotiation model presented in this work is based on the negotiation model proposed by Nash in [28]. It consists of the following steps:

1. At time step k, each subsystem sends to the remaining subsystems the values of $x_i(k),\ \eta_i(k)$.

2. With the information received, each subsystem solves the local optimization problem

$$\begin{aligned} & \max_{\widetilde{u}_i(k)} \sum_{r=1}^{M} w_r \log(\eta_r(k) - \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))) \\ \text{Subject to:} & \\ & \eta_r(k) > \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)),\ r = 1, \ldots, M \\ & \widetilde{u}_i(k) \in \Omega_i \end{aligned} \tag{3.11}$$

3. Let $\widetilde{u}_i^*(k)$ denote optimal control actions for subsystem i, $i = 1, \ldots, M$. If (3.11) is feasible, subsystem i selects the first control action of $\widetilde{u}_i^*(k)$ as a control action. Otherwise, subsystem i selects the first control action of $\widetilde{u}_i(k)$, where $\widetilde{u}_i(k)$ is the initial condition of subsystem i at time step k for solving (3.11).

4. Each subsystem updates its disagreement point. If (3.11) is feasible the update of the disagreement point of subsystem i is given by $\eta(k+1) = \eta_i(k) - \alpha[\eta_i(k) - \phi_i(\widetilde{u}(k))]$. Otherwise, the update of the disagreement point of subsystem i is given by $\eta(k+1) = \eta_i(k) + [\phi_i(\widetilde{u}(k)) - \eta_i(k)]$.

5. Each subsystem sends its updated control action and its updated disagreement point.

6. Go to step 1.

The initial condition for solving (3.11) at time step $k+1$ are given by the shifted control input $\widetilde{u}_{oi}(k+1) = [u_i^{*T}(k+1), \ldots, u_i^{*T}(k+N_\text{p}), 0]$, where the superscript $*$ denotes the optimal value of the control input. Negotiation model presented only considers the nonsymmetric case, but replacing (3.11) by

$$\begin{aligned} & \max_{\widetilde{u}_i(k)} \sum_{r=1}^{M} \log(\eta_r(k) - \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))) \\ \text{Subject to:} & \\ & \eta_r(k) > \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)),\ r = 1, \ldots, M \\ & \widetilde{u}_i(k) \in \Omega_i \end{aligned}$$

a distributed solution for symmetric DMPC games can be implemented. Note that although there is not an explicit negotiation process in proposed algorithm the cost function

$$J(\widetilde{u}(k)) = \sum_{r=1}^{M} w_r \log(\eta_r(k) - \sigma_r(\widetilde{u}_i(k), \widetilde{u}_{-i}(k)))$$

allows to every subsystem to have certain degree of coordination with the remaining subsystems. Thus, subsystem i is able to compute its optimal control inputs in a separated way from the information provided by the remaining subsystems. Furthermore, in comparison with the Lagrange multipliers based DMPC schemes, the proposed algorithm does not require an iterative process for computing the local control actions. This also allows to decrease the computational burden of the solution of the DMPC problem. In addition, using the definition of weighted hierarchy it is possible to analyze hierarchical MPC schemes as bargaining game, providing a general framework for applying MPC schemes to the large-scale systems control (this is not included in this work because is beyond of the scope of this work).

Furthermore, the bargaining process is formulated on the basis of how much each subsystem provide to the entire system performance and not in the concept of how each subsystem have to react against the decisions of the other subsystems. With this basis, the bargaining process allows to each subsystem to perceive a benefit from the cooperative behavior. Finally, although the formulation presented in this work comes from a special case in which the centralized cost function can be expressed as the sum of all local cost functions, this is not a requirement for the bargaining process. Only local functions that depend from decisions of the other subsystems is required. This makes more flexible the approach presented in this work than almost all the DMPC schemes presented in the literature.

In the next Section the convergence and the stability of the proposed DMPC scheme are analyzed.

3.4 Convergence and stability

Convergence and stability are the main properties of MPC schemes, specially for distributed and hierarchical MPC approaches. In this section the conditions for convergence and stability of the proposed method are established. Since the negotiation model presented in Section 3.3 is applicable to both symmetric and nonsymmetric DMPC games the conditions established for the convergence and the stability also are valid for both games. However, only the nonsymmetric case will be considered in this section.

3.4.1 Convergence

From the algorithm presented in Section 3.3, the convergence of the proposed DMPC method depends on the decision of each subsystem about to cooperate or to not. Assume that at time step $k = 0$ all subsystems decide to cooperate. Then $\eta_i(0) > \sigma_i(\widetilde{u}_i(0), \widetilde{u}_{-i}(0)),\ i = 1,\ldots,M$ and $\sum_{i=1}^{M} \eta_i(0) > \sum_{i=1}^{M} \sigma_i(\widetilde{u}_i(0), \widetilde{u}_{-i}(0))$. Moreover, (3.11) is feasible for all subsystems and the new value for the disagreement point is given by $\eta_i(1) = \eta_i(0) - \alpha[\eta_i(0) - \sigma_i(\widetilde{u}_i(0), \widetilde{u}_{-i}(0))],\ i = 1,\ldots,M$.

Let $C(k) \subset N$ denote the partition of N determined by the subsystems that decide to cooperate at time step k. Then, if at time step $k = 1,\ C(1) = N,\ \eta_i(1) < \eta_i(0)$ and therefore $\sigma_i(\widetilde{u}_i(0), \widetilde{u}_i(0)) > \sigma_i(\widetilde{u}_i(1), \widetilde{u}_i(1))$ for $i = 1,\ldots,M$. If the cooperative behavior remains for $k = 2,3,\ldots$, i.e., $C(k) = N,\ k = 1,2,\ldots$, then $\eta_i(0) > \eta_i(2) > \ldots$ Hence, $\sigma_i(\widetilde{u}_i(0), \widetilde{u}_i(0)) > \sigma_i(\widetilde{u}_i(1), \widetilde{u}_i(1)) > \ldots$ and $\sum_{i=1}^{M} \sigma_i(\widetilde{u}_i(0), \widetilde{u}_{-i}(0)) > \sum_{i=1}^{M} \sigma_i(\widetilde{u}_i(1), \widetilde{u}_{-i}(1)), \ldots$ Since the global cost function is equal to the sum of the local cost functions, i.e., $J_g(\tilde{x}(k), \widetilde{u}(k)) = \sum_{i=1}^{M} \phi_i(\widetilde{u}(k))$ and $\phi_i(\widetilde{u}(k)) = \sigma_i(\widetilde{u}_i(k), \widetilde{u}_{-i}(k))$ the global cost function is a decreasing function of k. Therefore the proposed algorithm converges if all subsystems decide to cooperate every time step k.

However, if at any time step k some subsystems decide not to cooperate the disagreement point of the noncooperating subsystems is increased in a factor $\sigma_j(\widetilde{u}_j(k), \widetilde{u}_{-j}(k))$ and $\widetilde{u}_j(k) = \widetilde{u}_j(k-1)$. With the increment of the value of the disagreement point the probability that the subsystems cooperate in the following time step $k+1$ increases (but is not guaranteed). Let $\eta_{i\max}$ denote the maximum value of the disagreement point of subsystem i. Then, if the subsystems does not cooperate at $k = 2,3,\ldots$ the value of $\eta_i(k)$ tends to $\eta_{i\max}$ as k tends to ∞. Moreover, $\widetilde{u}_i(k)$ remains constant. Assume that the system is stabilizable. Assume that Ω belongs to the set of control inputs of the stabilizable set. Then, although the subsystems does not cooperate the system states $x(k)$ converge to some stable trajectory. Therefore, the value of $J_g(\tilde{x}(k), \widetilde{u}(k))$ converges to a value and thus the proposed algorithm converge. Note that, if a subsystem does not cooperate does not imply that the control actions are unfeasible. The control actions of the noncooperating subsystems belongs to Ω_i but not to the set $\{u_i(k) \in \Omega_i\colon\ \eta_i(k) > \phi_i(k)\}$.

In the case that all subsystems do not cooperate at the beginning, the two cases analyzed before arise. Then the convergence of the algorithm is assured.

3.4.2 Stability

In order to demonstrate the stability of the closed-loop system two cases are considered:

1. All subsystems always cooperate.

2. Some subsystems do not cooperate but few time steps ahead start to cooperate

In the first case, let $V(\tilde{x}(k),\widetilde{u}(k)) = J_g(\tilde{x}(k),\widetilde{u}(k))$ be the Lyapunov function of the system. Then we have to prove that $V(\tilde{x}(k),\widetilde{u}(k))$ is positive for $(\tilde{x}(k),\widetilde{u}(k)) \neq 0$ and equals to 0 if $(\tilde{x}(k),\widetilde{u}(k)) = (0,0)$, and that $V(\tilde{x}(k),\widetilde{u}(k)) - V(\tilde{x}(k-1),\widetilde{u}(k-1)) < 0$. Since $\phi_i(\widetilde{u}(k))$ is a quadratic positive convex function, and since $J_g(\tilde{x}(k),\widetilde{u}(k)) = \sum_{i=1}^{M} \phi_i(\widetilde{u}(k))$ the Lyapunov candidate function $V(\tilde{x}(k),\widetilde{u}(k))$ is positive for $(\tilde{x}(k),\widetilde{u}(k)) \neq 0$ and equals to 0 if $(\tilde{x}(k),\widetilde{u}(k)) = (0,0)$.

Assume that all subsystem always cooperate. Then, $\eta_i(k-1) > \eta_i(k)$, $\sigma_i(\widetilde{u}_i(k-1),\widetilde{u}_i(k-1)) > \sigma_i(\widetilde{u}_i(k),\widetilde{u}_i(k))$, and $\sum_{i=1}^{M} \sigma_i(\widetilde{u}_i(k-1),\widetilde{u}_{-i}(k-1)) > \sum_{i=1}^{M} \sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k))$ for all time step k. Since $\eta_i(k) = \eta_i(k-1) - \alpha[\eta_i(k-1) - \sigma_i(\widetilde{u}_i(k-1),\widetilde{u}_{-i}(k-1))]$, $i = 1,\ldots,M$ and $\eta_i(k) > \sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k))$ for all cooperating subsystems $\sum_{i=1}^{M} \sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k)) - \sum_{i=1}^{M} \sigma_i(\widetilde{u}_i(k-1),\widetilde{u}_{-i}(k-1)) < -\sum_{i=1}^{M} \alpha(\eta_i(k-1) - \sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k)))$. Recall that $J_g(\tilde{x}(k),\widetilde{u}(k)) = \sum_{i=1}^{M} \phi_i(\widetilde{u}(k))$ and $\phi_i(\widetilde{u}(k)) = \sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k))$. So, $V(\tilde{x}(k),\widetilde{u}(k)) = \sum_{i=1}^{M} \sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k))$ and $V(\tilde{x}(k-1),\widetilde{u}(k-1)) - V(\tilde{x}(k),\widetilde{u}(k)) < -\sum_{i=1}^{M} \alpha(\eta_i(k-1) - \sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k)))$. Thus, $V(\tilde{x}(k),\widetilde{u}(k))$ is a positive function bounded below by 0 which decreases as $\sum_{i=1}^{M} \alpha(\eta_i(k-1) - \sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k)))$. Therefore, $V(\tilde{x}(k),\widetilde{u}(k))$ tends to 0 as k tends to ∞. Hence the closed-loop system is stable if all subsystems always decide to cooperate.

If some subsystems decide not to cooperate at any time step k ($NC(k) \neq \varnothing$), then $V(\tilde{x}(k),\widetilde{u}(k)) = J_{gC}(\tilde{x}(k),\widetilde{u}(k)) + J_{gNC}(\tilde{x}(k),\widetilde{u}(k))$, where the index C is related with the subsystems that decide to cooperate, and the index NC is related with the subsystems that decide not to cooperate. In this case $J_{gC}(\tilde{x}(k),\widetilde{u}(k))$ is a decreasing function tending to 0 as k tends to ∞, but $J_{gNC}(\tilde{x}(k),\widetilde{u}(k))$ is a function tending to some value as k tends to ∞ (this value is given by the behavior of the states). Therefore it is not possible to assure that $V(\tilde{x}(k),\widetilde{u}(k))$ is a decreasing function, but it is possible to guarantee that $V(\tilde{x}(k),\widetilde{u}(k))$ converges to certain finite value. However, when the subsystems that decide not to cooperate begin to cooperate again, $V(\tilde{x}(k),\widetilde{u}(k))$ behaves as mentioned in the case in which all subsystems decide to cooperate. Then the stability of the closed-loop system is also assured.

The case where all/some subsystems always decide not to cooperate is not included because the global cost function $J_g(\tilde{x}(k),\widetilde{u}(k))$ tends to some value (under the assumptions that the system is stabilizable), but it does not tends to 0. In this case, only, it is possible to conclude that the states of the system converges to a neighborhood at the origin.

3.5 Simulation results

In this section simulation results are presented for both symmetric and nonsymmetric DMPC games. For illustrating a case in which the proposed DMPC satisfies the symmetry conditions, the quadruple tank process presented in [45] is used. In the nonsymmetric case, the hydro-power valley (HPV) described in [31] is considered.

3.5.1 Symmetric distributed model predictive control game

The four-tank plant is a laboratory plant that has been designed to test control techniques using industrial instrumentation and control systems. The plant consists of a hydraulic process of four interconnected tanks inspired by the educational quadruple-tank process proposed by [45]. The process constitutes a simple multivariable system with highly coupled nonlinear dynamics that can exhibit transmission zero dynamics. Figure 3.1 shows a schematic diagram of the quadruple tank process.

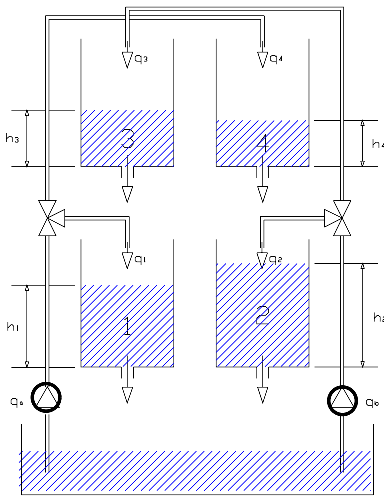


Figure 3.1: Johansson's quadruple-tank process diagram.

A continuous-time state-space model of the quadruple-tank process system can be derived from first principles [45] to result in:

$$\frac{dh_1(t)}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1(t)} + \frac{a_3}{A_1}\sqrt{2gh_3(t)} + \frac{\gamma_a}{A_1}q_a(t) \tag{3.12}$$

$$\frac{dh_2(t)}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2(t)} + \frac{a_4}{A_2}\sqrt{2gh_4(t)} + \frac{\gamma_b}{A_2}q_b(t) \tag{3.13}$$

$$\frac{dh_3(t)}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3(t)} + \frac{(1-\gamma_b)}{A_3}q_b(t) \tag{3.14}$$

$$\frac{dh_4(t)}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4(t)} + \frac{(1-\gamma_a)}{A_4}q_a(t) \tag{3.15}$$

where $h_i(t)$, A_i and a_i with $i \in \{1,2,3,4\}$ refer to the level, cross section and the discharge constant of tank i, respectively; $q_j(t)$ and γ_j with $j \in \{a,b\}$ denote the flow and the ratio of the three-way valve of pump j, respectively and g is the gravitational acceleration.

The plant parameters are shown in Table 3.2.

Linearizing the model at an operating point given by the equilibrium levels and flows shown in Table 3.2 and defining the deviation variables $x_i(t) = h_i(t) - h_i^0(t)$, $u_j(t) = q_j(t) - q_j^0$ we obtain the continuous-time linear model

$$\begin{aligned} \frac{dx}{dt} &= A_c x(t) + B_c u(t). \\ y(t) &= Cx(t) \end{aligned} \tag{3.16}$$

	value	unit	description
h_{max}	1.36	m	Maximum level in all cases
h_{min}	0.2	m	Minimum level in all cases
q_{max}	3.26	m^3/h	Maximum flow of q_a and q_b
q_{min}	0	m^3/h	Minimum flow of q_a and q_b
a_1, a_2 a_3, a_4	1.31e-4	m^2	Discharge constant of the tanks
A_1, A_2 A_3, A_4	0.06	m^2	Cross-section of the tanks
γ_a, γ_b	0.3		Parameter of the 3-way valve of q_a and q_b
h_1^0	0.6042	m	Linearization level of tank 1
h_2^0	0.6042	m	Linearization level of tank 2
h_3^0	0.296	m	Linearization level of tank 3
h_4^0	0.296	m	Linearization level of tank 4
q_a^0, q_b^0	1.63	m^3/h	Linearization flow of q_a and q_b
T_s	5	s	Sample time

Table 3.2: Parameters of the plant

where $x(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]^T$, $u(t) = [u_1(t), u_2(t)]^T$, $y(t) = [x_1(t), x_2(t)]^T$, and

$$A_c = \begin{bmatrix} \frac{-1}{\tau_1} & 0 & \frac{1}{\tau_3} & 0 \\ 0 & \frac{-1}{\tau_2} & 0 & \frac{1}{\tau_4} \\ 0 & 0 & \frac{-1}{\tau_3} & 0 \\ 0 & 0 & 0 & \frac{-1}{\tau_4} \end{bmatrix}, \; B_c = \begin{bmatrix} \frac{\gamma_a}{A_1} & 0 \\ 0 & \frac{\gamma_b}{A_2} \\ 0 & \frac{(1-\gamma_b)}{A_3} \\ \frac{(1-\gamma_a)}{A_4} & 0 \end{bmatrix}, \; C_c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

with $\tau_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}} \geq 0$ the time constant of tank i.

Similar than in [37], in this work the whole system is divided into two coupled subsystems as follows: The subsystem 1 consists of tanks 1 and 3 while subsystem 2 consists of tanks 2 and 4, that is, the subsystem 1 is characterized by the state $\mathbf{x}(t)_1 = [x_1(t), x_3(t)]^T$ and its output $\mathbf{y}_1(t)$ is $x_1(t)$ while the state of the subsystem 2 is $\mathbf{x}_2(t) = [x_2(t), x_4(t)]^T$ and its output $\mathbf{y}_2(t) = x_2(t)$. The continuous-time models of subsystems 1 and 2 are given by

$$\begin{aligned} \frac{d\mathbf{x}_1(t)}{dt} &= A_{c1}\mathbf{x}_1(t) + B_{c1}u(t) \\ \mathbf{y}_1(t) &= C_{c1}\mathbf{x}_1(t) \end{aligned} \tag{3.17}$$

and by

$$\begin{aligned} \frac{d\mathbf{x}_2(t)}{dt} &= A_{c2}\mathbf{x}_2(t) + B_{c2}u(t) \\ \mathbf{y}_2(t) &= C_{c2}\mathbf{x}_2(t) \end{aligned} \tag{3.18}$$

respectively, where A_{c1}, B_{c1} , C_{c1}, A_{c2}, B_{c2} and C_{c2} comes from A_c, B_c, C_c. Note that the subsystems considered are coupled only through the inputs. The corresponding discrete-time model of each subsystem is derived from the previous ones by means of the Tustin method. These will be denoted as follows:

$$\begin{aligned} \mathbf{x}_1(k+1) &= A_{d1}\mathbf{x}_1(k) + B_{d1}u(k) \\ \mathbf{y}_1(k) &= C_{d1}\mathbf{x}_1(k) \end{aligned} \tag{3.19}$$

$$\begin{aligned} \mathbf{x}_2(k+1) &= A_{d2}\mathbf{x}_2(k) + B_{d2}u(k) \\ \mathbf{y}_2(k) &= C_{d2}\mathbf{x}_2(k) \end{aligned} \tag{3.20}$$

In order to test the proposed DMPC scheme, we define a simulation experiment in which the control objective is to follow the changes in the reference values of tanks 1 and 2 by manipulating the flows q_a and q_b. The changes made in the reference values were:

1. The reference values of thanks 1 and 2 were set at h_1^0 and h_2^0 respectively.
2. At 3000s a change of $-0.1h_1^0$ in the reference value of the tank 1 and a change of $0.1h_2^0$ in the reference value of tank 2 were made.
3. At 6000s the reference values was returned to h_1^0 and h_2^0 respectively.
4. At 9000s a 30% increase in reference values of thanks 1 and 2 corresponding to the 30% of h_1^0 and h_2^0 respectively were made.

For the design of the DMPC scheme, a local quadratic cost function

$$\begin{aligned} L_i(\widetilde{x}_i(k), \widetilde{u}_i(k)) = &\sum_{t=0}^{N_\mathrm{p}-1} \left[\mathbf{x}_i^T(k+t|k) Q_i \mathbf{x}_i(k+t|k)\right] + \sum_{t=0}^{N_\mathrm{u}} \left[u_i^T(k+t) R_i u_i(k+t)\right] \\ &+ \mathbf{x}_i^T(k+N_\mathrm{p}|k) P_i \mathbf{x}_i(k+N_\mathrm{p}|k) \end{aligned} \tag{3.21}$$

$i = 1,2$ is used to measure the performance of each subsystem. In (3.21), $\mathbf{x}_i(k+t|k)$ denotes the predicted value of $\mathbf{x}_i$ at time step $k+t$ given the conditions at time step k, $u_i(k+t)$ denotes the control input u_i at time step $k+t$, $\widetilde{x}_i(k) = [\mathbf{x}_i^T(k|k), \ldots, \mathbf{x}_i^T(k+N_\mathrm{p}|k)]^T$, $\widetilde{u}_i(k) = [u_i^T(k), \ldots, u_i^T(k+N_\mathrm{u}), \ldots, u_i^T(k+N_\mathrm{p})]^T$, where $\mathbf{x}_i(k|k) = \mathbf{x}_i(k)$, and $u_i(k+t) = u_i(k+N_\mathrm{u})$, for $t = N_\mathrm{u}+1, \ldots, N_\mathrm{p}-1$, Q_i, R_i are diagonal matrices with positive diagonal elements, and P_i being the solution of the Lyapunov equation

$$A_{di}^T P_i A_{di} - P_i = -Q_i$$

Substituting the expression for $\mathbf{x}_i(k+t|k)$ into (3.21), and by using the control horizon constraint $u_i(k+t) = u_i(k+N_\mathrm{u})$, for $t = N_\mathrm{u}+1, \ldots, N_\mathrm{p}-1$, the function $L_i(\widetilde{x}_i(k), \widetilde{u}_i(k))$ can be expressed as a quadratic function $\phi_i(\widetilde{u}(k))$, $\mathbf{x}_i(k)$ being the value of the state vector at time step k of subsystem i. Thus, the cost function of each subsystem becomes

$$\phi_i(\widetilde{u}(k)) = \widetilde{u}^T(k) Q_{uui} \widetilde{u}(k) + 2\mathbf{x}_i^T(k) Q_{xui} \widetilde{u}(k) + \mathbf{x}_i^T(k) Q_{xxi} \mathbf{x}_i(k) \tag{3.22}$$

where $Q_{uui} \geq 0$, for $i = 1, \ldots, M$. Then, we have a game $G_\mathrm{tank} = \{N, \{\phi_i(\widetilde{u}(k)\}_{i\in N}, \{\Omega_i\}_{i\in N}\}$, with $N = \{1,2\}$, where all subsystems have the same goal: to compute the optimal control inputs such that the global performance of the system is maximized, i.e., the reference tracking is achieved by each subsystems. In the game G_tank, Ω_i is given by the constraints in Table 3.2.

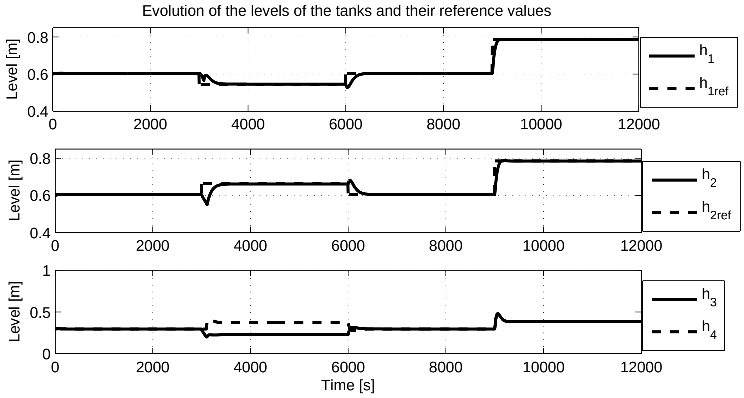


Figure 3.2: Evolution of the levels of the tanks.

Figure 3.2 shows the behavior of the levels of the liquid inside of the tanks when the control actions are computed as the solution of a DMPC game. From this figure it is possible to conclude that the output levels are regulated to the desired values despite of the changes of the set points.

Figure 3.3 shows the evolution of the control inputs, the disagreement point, and the value of the cost function of each subsystem. From this figure we have:

1. The disagreement points satisfies $\eta_1(k) = \eta_2(k)$ for all time step k.
2. The solution of the DMPC game associated with the four-tank system satisfies $\varphi_1(\Upsilon, \eta(k)) = \varphi_2(\Upsilon, \eta(k))$, for all k.
3. $u_1(k) \in (\Omega_1 \cap \Omega_2)$ and $u_2(k) \in (\Omega_1 \cap \Omega_2)$ for all k. Then $\widetilde{\phi}(\widetilde{u}(k)) \in \Upsilon$.

So, we can conclude that under the conditions presented in this paper the DMPC problem is a symmetric DMPC game. Moreover, in this figure it is also shown that the values of the states are not required to be the same to maintain the symmetry of the game. The symmetry of the game is given by the symmetry of the system.

3.5.2 Nonsymmetric distributed model predictive control game

Consider the HPV shown in Figure 3.4. This HPV is composed by three lakes ($L_m,\ m = 1,2,3$), two of them ($L_1,\ L_2$) connected by a duct (U_1), and six dams ($D_j,\ j = 1,\dots,6$), each one of them equipped with a turbine for electric power production. The dams are located into a river, dividing it in six reaches (R_j). Reaches $R_1,\ R_2$ are connected with lake L_1 through a turbine-pump (C_1) and a turbine (T_1) respectively. Moreover, reaches $R_4,\ R_5$ are connected with lake L_3 by a turbine-pump (C_2) and a turbine (T_2) respectively. Turbines T_1, T_2 also are used for electric power generation, and turbine-pump devices $C_1,\ C_2$ are used to produce electric power (in turbine mode) and to pump water from reaches $R_1,\ R_4$ in order to regulate the level of the lakes $L_2,\ L_3$ respectively. Note that in pump mode $C_1,\ C_2$ consume electric power. Furthermore, reaches $R_1,\ R_3$ are fed by the river flow q_{in} and the tributary flow $q_{tributary}$ respectively.

A model suitable for control purposes for the HPV of Figure 3.4 is derived in [31]. This model is based on the following assumptions:

† The ducts are connected at the bottom of the lakes (or to the bottom of the river bed).

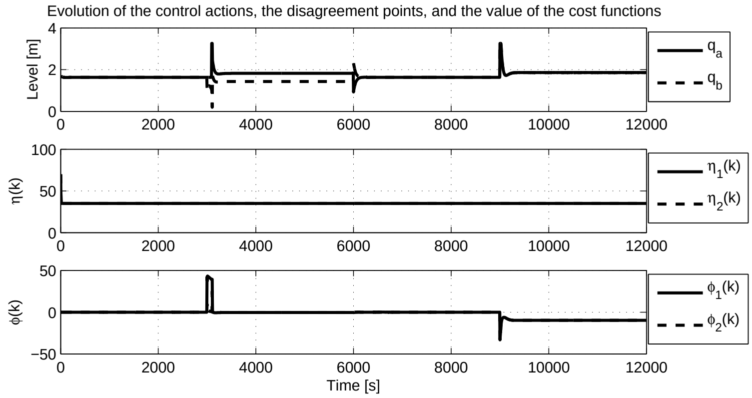


Figure 3.3: Evolution of the control actions, disagreement points, and value of the cost function of each subsystem

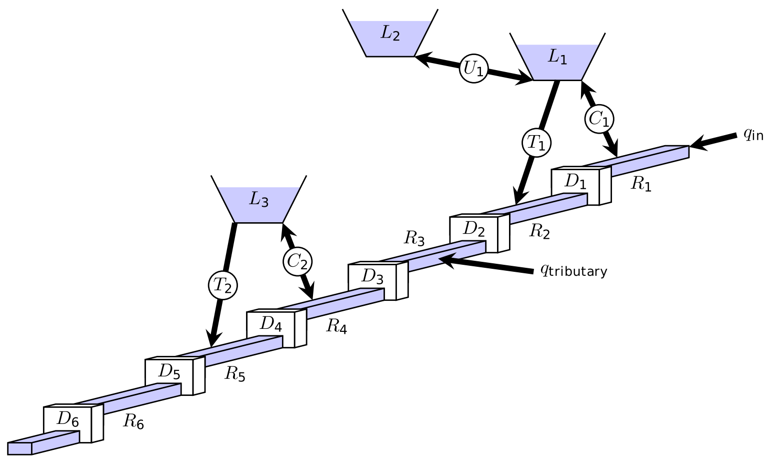


Figure 3.4: HPV

† The cross section of the reaches and of the lakes is rectangular.

† The width of the reaches varies linearly along the reaches.

† The river bed slope is constant along every reach

Based on these assumptions, the nonlinear, first-order Saint-Venant partial differential equations represent the state of the art for modeling one-dimensional river hydraulics with constant fluid density [46]. In this equation the hydraulic state of the river is described by two variables: the water depth $h(t,z)$ and the discharge $q(t,z)$, both varying as a function of space z and time t. Thus, the dynamics of each reach are given by [31, 46, 47, 48]

$$\frac{\partial q}{\partial z} + \frac{\partial s}{\partial t} = 0 \tag{3.23}$$
$$\frac{1}{g}\frac{\partial}{\partial t}\left(\frac{q}{s}\right) + \frac{1}{2g}\frac{\partial}{\partial z}\left(\frac{q^2}{s^2}\right) + \frac{\partial h}{\partial z} + I_f - I_o = 0$$

In (3.23), $q = q(t,z),\ s = s(t,z),\ h = h(t,z),\ I_f = I_f(t,z),\ I_o = I_o(t,z)$, where $s(t,z)$ is the wetted surface, $I_f(t,z)$ is the friction slope, $I_o(t,z)$ is the river bed slope, and g is the gravitational acceleration. Since the cross section of the reaches and of the lakes is assumed rectangular the wetted surface and the friction slope are given by (3.24) and (3.25) respectively [31].

$$s(t,z) = w(z)h(t,z) \tag{3.24}$$

$$If(t,z) = \frac{q^2(t,z)(w(z)+2h(t,z))^{\frac{4}{3}}}{k_{str}^2(w(z)h(t,z))^{\frac{10}{3}}} \tag{3.25}$$

where $w(z)$ is the river width, and k_{str} is the Gauckler-Manning-Strickler coefficient. For modeling the lakes, the duct, the turbines, and the turbine-pumps elements (3.26)-(3.29) are used [31].

$$\frac{\partial h(t)}{\partial t} = \frac{q_{in}(t) - q_{out}(t)}{S} \tag{3.26}$$

$$q_{U1}(t) = S_{U1}\mathrm{sign}(H(t))\sqrt{2g|H(t)|} \tag{3.27}$$

$$p_t(t) = k_t q_t(t)\Delta h_t(t) \tag{3.28}$$

$$p_C(t) = k_C(q_C(t))q_C(t)\Delta h_C(t) \tag{3.29}$$

where S is the surface area of the lake, S_{U1} is the section of the duct, k_t is the turbine coefficient, $q_{in}(t),\ q_{out}(t)$, are the input and output flows of the lakes respectively, $q_t(t)$ is the turbine discharge, $\Delta h_t(t),\ \Delta h_C(t)$ are the heads of the turbine and the turbine-pump respectively,

$$k_C(q_C(t)) = \left\{ \begin{array}{l} k_{tc} \text{ if } q_C(t) \geq 0 \\ k_{pC} \text{ if } q_C(t) < 0 \end{array} \right\}$$

is the turbine-pump coefficient, and $H(t) = h_{L2}(t) - h_{L1}(t) + h_{U1}$, with $h_{L1}(t),\ h_{L2}(t)$ the levels of the lakes 1 and 2 respectively, and h_{U1} the height difference of the duct.

Although (3.23)-(3.29) describe the dynamic behavior of the HPV. This model is unsuitable for control purposes. Then in order to obtain a model suitable for control purposes, a spatial discretization of (3.23) is required. The expressions of the resulting model are given in [31]. Let $x(t),\ u(t)$ denote the states and the inputs of the system. Then

$$u(t) = [q_{T1}(t), q_{C1}(t), q_{T2}(t), q_{C2}(t), q_{R1}(t), q_{R2}(t), q_{R3}(t), q_{R4}(t), q_{R5}(t), q_{R6}(t)]^T$$
$$x(t) = [h_{Lm}^T(t), q_{R1}^T(t), h_{R1}^T(t), q_{R2}^T(t), h_{R2}^T(t), q_{R3}^T(t), h_{R3}^T(t), q_{R4}^T(t), h_{R4}^T(t), q_{R5}^T(t), h_{R5}^T(t), q_{R6}^T(t), h_{R6}^T(t)]^T$$

with $q_{Rl} = [q_1(t), \ldots, q_{N_x}(t)]$, and $h_{Rl} = [h_1(t), \ldots, h_{N_x+1}(t)]$ the flows and the levels at each spatial partition of reach $R_l,\ l = 1, \ldots, 6,\ q_{Tp},\ q_{Cp},\ q_{Rl},\ p = 1,2$ the flows of the turbines, the turbine-pumps, and the turbines at the reaches, and N_x being the number of partitions. This model is used for implementing in simulation a DMPC scheme for the control of the HPV.

The DMPC scheme proposed is designed considering the power tracking scenario proposed in [31]. In this scenario, power output of the system should follow a given reference while keeping the water levels in the lakes and at the dams as constant as possible. So, the global cost function considered for the DMPC is composed by two terms: the first term penalizes the 1-norm of the power tracking error, and the second term penalizes the 2-norm of the deviations of the levels in the lakes and in the dams of their steady state values. Thus, the centralized MPC problem is formulated as follows

[31]

$$
\begin{aligned}
&\min_{u(t)} \int_0^T \lambda |p_r(t) - p(x(t),u(t))| dt + \int_0^T (h(t)-h_{ss})^T Q (h(t)-hss) \\
&\text{Subject to:} \\
&\dot{x}(t) = f(x(t),u(t)) \\
&u(t) \in C
\end{aligned} \tag{3.30}
$$

where $\dot{x}(t) = \frac{dx(t)}{dt}$, T is the prediction horizon, $\lambda > 0$, $Q > 0$ are diagonal matrices, $p_r(t), p(t)$ are the power reference and the power produced by the HPV respectively, h_{ss} is the vector of the steady state levels, $f(.)$ is a function representing the HPV dynamics, and C is the feasible set composed by the constraints on $u(t)$ and $x(t)$. The power reference to be followed by the entire system is known 24 hours in advance and the inputs of the system can be changed every 30 minutes.

Let T_s denote the sample time. Then the HPV model (3.23)-(3.29) can be expressed as a linear system as

$$
\begin{aligned}
x(k+1) &= A_d x(k) + B_d u_k \\
y(k) &= C_d x(k) + D_d u(k)
\end{aligned} \tag{3.31}
$$

where A_d, B_d, C_d, D_d are the matrices resulting of the linearization of (3.23)-(3.29), and $y(k)$ is the output of the system, i.e., $y(k) = [p(k), h_D^T(k)]^T$, with $h_D(k) = [h_{D1N_x}, h_{D2N_x}, h_{D3N_x}, h_{D4N_x}, h_{D5N_x}, h_{D6N_x}]$ the levels at the dams. Note that the power produced by the HPV is piecewise defined respect to $u(k)$ due to the turbine-pump elements. In order to overcome this issue in the linearization, constants k_{des1}, k_{des2} was introduced, virtual inputs $\bar{u}_1(k) \in [-q_{pC1}, q_{tC1}], \bar{u}_2(k) \in [-q_{pC2}, q_{tC2}]$ was considered, and a gain compensation

$$
u_p(k) = \left\{ \begin{array}{l} \frac{k_{desp}}{k_{tCp}} \bar{u}_p(k) \text{ if } \bar{u}_p(k) \geq 0 \\ \frac{k_{desp}}{k_{pCp}} \bar{u}_p(k) \text{ if } \bar{u}_p(k) < 0 \end{array} \right\}
$$

was proposed, where $q_{pC1}, q_{pC2}, q_{tC1}, q_{tC2}$ are the maximum pumped flows and maximum turbined flows for the turbine-pump elements C_1, C_2 respectively, $p = 1, 2$ (the values of $q_{pC1}, q_{pC2}, q_{tC1}, q_{tC2}$ are given in [31]).

Note that optimization problem (3.30) can be written as

$$
\begin{aligned}
&\min_{\widetilde{u}(k)} \gamma |\widetilde{p}_r(k) - \widetilde{y}_p(\widetilde{u}(k))| + \widetilde{u}^T(k) H \widetilde{u}(k) + 2F\widetilde{u}(k) \\
&\text{Subject to:} \\
&\widetilde{u}(k) \in \Omega \\
&u(k+\nu) = u(k+N_\text{u}),\ \forall N_\text{u} < \nu < N_\text{p} - 1
\end{aligned} \tag{3.32}
$$

where $\widetilde{p}_r(k) = [p_r(k), \ldots, p_r(k+N_\text{p})]$, $\widetilde{y}_p(\widetilde{u}(k)) = [p(x(k),u(k)), \ldots, p(x(k),u(k+N_\text{p}-1))]$, $H = \bar{B}_d^T \bar{Q} \bar{B}_d$, $F = x^T(k) \bar{A}_d^T \bar{Q} \bar{B}_d$, and Ω is the feasible set composed by the input constraints and the mapping using (3.31) of the state constraints to input constraints, with $\bar{A}_d, \bar{B}_d$ the resulting matrices by the prediction of $h_D(k)$ along N_p. From [31], it is possible to divide the HPV of Figure 3.4 into 8 subsystems:

† Subsystem 1: lakes 1 and 2, turbine $T1$, and turbine-pump $C1$.

- † Subsystem 2: lake 3, turbine $T2$, and turbine-pump $C2$.
- † Subsystems 3-8: reaches R_1 to R_6 respectively.

Let

$$\sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k)) = \gamma|\widetilde{p}_r(k) - \widetilde{y}_p(\widetilde{u}_i(k),\widetilde{u}_{-i}(k))| + [\widetilde{u}_i(k),\widetilde{u}_{-i}(k)]^T \bar{H}_i[\widetilde{u}_i(k),\widetilde{u}_{-i}(k)] + 2\bar{F}_i[\widetilde{u}_i(k),\widetilde{u}_{-i}(k)]$$

where $\bar{H}_i, \bar{F}_i$ are the resulting matrices of the permutation of the rows and columns of H, F respectively. (the state dependence of $\sigma_i(.)$ was omitted for notational convenience). From [31], the state and input constraints are time independent an only establishes lower and upper boundaries to the states and inputs. So, they are independent for each subsystems, i.e., there is not coupled constraints. Then, for the control of the HPV we have a game $G_{HPV} = \{N, \{\sigma_i(\widetilde{u}_i(k),\widetilde{u}_{-i}(k))\}_{i\in N}, \{\Omega_i\}_{i\in N}\}$, with $N = \{1,\ldots,8\}$, in which all subsystems have the same goal: to minimize the power tracking error keeping the levels in the lakes and at the dams as close as possible to their steady state values. Hence, the game G_{HPV} can be analyzed and solved as a discrete-time dynamic bargaining game $\{(\Upsilon, \eta(k))\}_{k=0}^{\infty}$.

Note that the power produced by the HPV at time step k is equals to the sum of the powers generated by all subsystems, and assuming that each subsystem communicates the value of the states and inputs to the remaining subsystems, each subsystem is able to compute the power produced by the other subsystems. Hence, the term $\gamma|\widetilde{p}_r(k) - \widetilde{y}_p(\widetilde{u}_i(k),\widetilde{u}_{-i}(k))|$ is reduced to compute the power contribution of subsystem i given the power produced by the remaining subsystems.

Based on the formulation presented in this section, a closed-loop simulation of the HPV described of Figure 3.4 was performed along 24 hours (simulation time). In this simulation, $k_{des1} = \frac{3}{4}(k_{tC1} + k_{pC1})$, $k_{des2} = \frac{3}{4}(k_{tC2} + k_{pC2})$, $T_s = 1800s$ (30 minutes), $N_\mathrm{p} = 48$ (corresponding to a day), $N_\mathrm{u} = 32$, $w_{1,2} = \frac{0.4}{2}$, $w_{3-8} = \frac{0.6}{6}$ (the weights of subsystems 1 to 8), $d(0) = 1 * 10^5$, $\gamma = 50$, $Q = I$ (I being the identity matrix), and the lower and upper values of the inputs and the states, and the parameters of the model (3.23)-(3.29) were taken as the proposed in [31].

Figure 3.5 shows the comparison between the power produced by the HPV and the power reference when the proposed DMPC scheme computes the inputs of each subsystem. In this Figure it is shown that the power produced by the HPV follows the power reference, satisfying one of the objectives proposed for the control scheme.

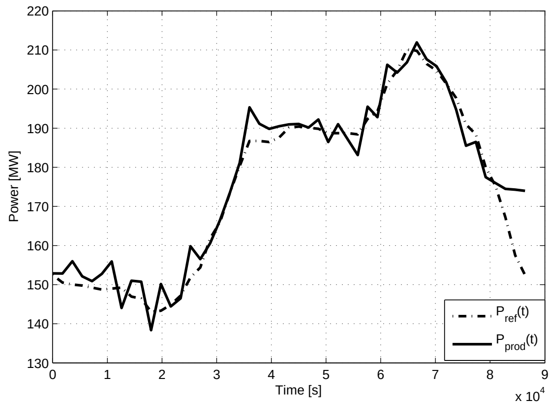


Figure 3.5: Comparison between the power produced by the HPV with the power reference, when the proposed game-theory-based DMPC is used for computing the inputs of the subsystems

In order to maintain the power demand, the levels of the reaches and the lakes should be modified. In Figure 3.3 the behavior of the levels is presented. Although the levels of the lakes have larger variations (see first panel of Figure 3.3) than the dams levels (see second panel of Figure 3.6), the second objective proposed for the control scheme is partially satisfied, because the levels at the dams are maintained as constant as possible. If it is considered that the reaches also can be used for maritime traffic, then maintaining the levels of the reaches such a traffic can be assured.

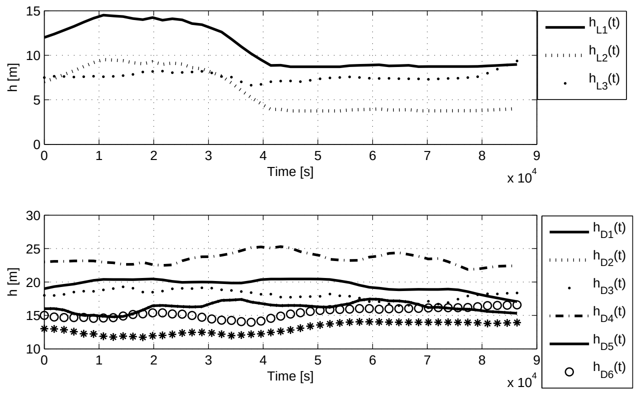


Figure 3.6: Behavior of the levels in the lakes (first panel) and the levels at the dams (second panel) of the HPV.

Moreover, despite of the lost of performance associated with the large excursions of the levels of the lakes, all the control inputs applied to the subsystems are inside of the defined level for them (see Figure 3.7).

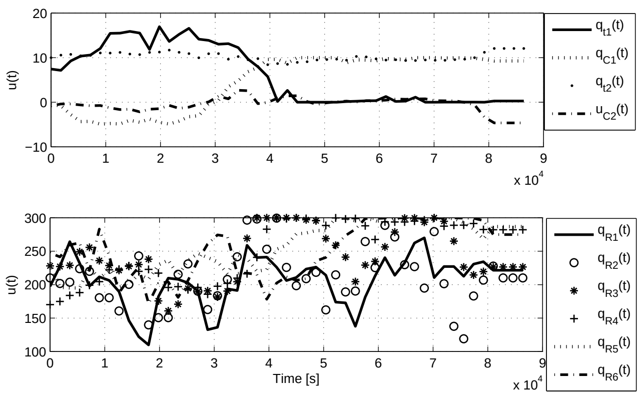


Figure 3.7: Behavior of the applied control actions to each subsystem

Finally, in Figure 3.8 the evolution of the disagreement points is presented. In this Figure, the disagreement starts at the same point but as they are evolving each subsystem has their own value indicating the nonsymmetry of the game G_{HPV} (see Figure 3.9 for a zoom in showing the different values of the disagreement points).

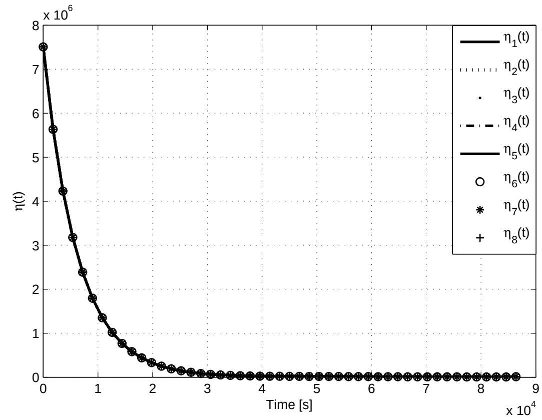


Figure 3.8: Evolution of the disagreement points

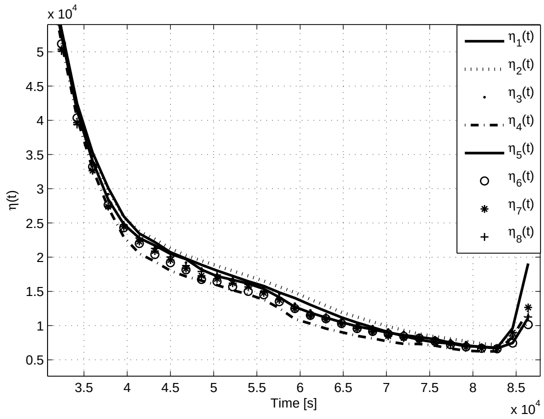


Figure 3.9: Zoom in of the evolution of the disagreement points

3.6 Concluding remarks

In this work the DMPC problem was characterized as a bargaining game by the axiomatic bargaining game theory proposed by Nash for such games. From the axiomatic theory, conditions for the symmetry and nonsymmetry of the game associated with the DMPC problem were established. In order to derive these conditions several concepts of the original theory should be redefined in order to include the time evolution of the games. The conditions established allowed to conclude that in real large-scale systems the symmetry conditions are heavily restrictive, because real large-scale systems have several elements with different state equations making difficult the achievement of the symmetry. On the other hand, the characterization of the DMPC problem as a nonsymmetric game allowed to conclude that the mathematical framework defined for nonsymmetric DMPC games can be also applied for hierarchical MPC schemes, providing a mathematical framework for MPC applied to the large-scale systems control.

Moreover, a negotiation model including both symmetric and nonsymmetric DMPC games was proposed. This algorithm is based on the transformation of the bargaining game in an equivalent noncooperative game, and solve the equivalent noncooperative game. The transformation allowed to reduce the computational burden associated with the solution of the DMPC problem because it is not required an iterative procedure for jointly compute the optimal control action applied to each subsystem, which is the main characteristic of the widely used Lagrange multipliers based DMPC methods. The convergence and stability of the proposed control scheme were also discussed.

Finally, two application cases were presented: the quadruple tank process for illustrating a possible situation in which the DMPC problem can be solved as a symmetric game, and the hydro-power valley proposed in [31] for illustrating a situation in which the DMPC problem can be solved as a nonsymmetric game. Both results shows the capabilities of the solution of a DMPC problem as a bargaining game.

Bibliography

[1] R. Negenborn, *Multi-Agent Model Predictive Control with Applications to Power Networks*. PhD thesis, Delft University of Technology, Delft, The Netherlands, Dec. 2007.

[2] R. Negenborn, B. De Schutter, and J. Hellendoorn, “Multi-agent model predictive control for transportation networks: Serial versus parallel schemes,” *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 353–366, 2008.

[3] S. Talukdar, D. Jia, P. Hines, and B. H. Krogh, “Distributed model predictive control for the mitigation of cascading failures,” *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pp. 4440–4445, 2005.

[4] F. Wang and I. Cameron, “A multi-form modelling approach to the dynamics and control of drum granulation processes,” *Powder Technology*, vol. 179, no. 1-2, pp. 2–11, 2007.

[5] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, “Distributed Model Predictive Control,” *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[6] D. Jia and B. H. Krogh, “Distributed Model Predictive Control,” *Proceedings of the 2001 American Control Conference*, pp. 2767–2772, 2001.

[7] I. Necoara, D. Doan, and J. A. K. Suykens, “Application of the proximal center decomposition method to distributed model predictive control,” *Proceedings of the 2008 IEEE Conference on Decision and Control*, pp. 2900–2905, 2008.

[8] W. Dunbar and R. Murray, “Distributed receding horizon control for multi-vehicle formation stabilization,” *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[9] D. Doan, T. Keviczky, I. Necoara, and M. Diehl, “A jacobi algorithm for distributed model predictive control of dynamically coupled systems,” Tech. Rep. arXiv:0809.3647, Sep 2008. Comments: Shorter version submitted to 2009 American Control Conference.

[10] D. Jia and B. H. Krogh, “Min-max feedback model predictive control for distributed control with communication,” *Proceedings of the 2002 American Control Conference*, pp. 4507–4512, 2002.

[11] A. Rantzer, “Linear quadratic team theory revisited,” *Proceedings of the 2006 American Control Conference*, 2006.

[12] A. Rantzer, “Using game theory for distributed control engineering,” in *3rd World Congress of the Game Theory Society Games*, 2008.

[13] A. Rantzer, “Dynamic dual decomposition for distributed control,” *American Control Conference 2009*, 2009.

[14] A. Venkat, I. Hiskens, J. Rawlings, and S. Wright, “Distributed MPC strategies for automatic generation control,” in *Proceedings of the IFAC Symposium on Power Plants and Power Systems Control*, pp. 25–28, 2008.

[15] A. Venkat, J. Rawlings, and S. Wright, “Stability and optimality of distributed, linear model predictive control. Part I: State feedback,” *Texas-Wisconsin Modeling and Control Consortium Technical Report*, vol. 3, 2006.

[16] A. N. Venkat, J. B. Rawlings, and S. J. Wright, “Implementable distributed model predictive control with guaranteed performance properties,” *Proceedings of the 2006 American Control Conference*, pp. 613–618, 2006.

[17] W. Dunbar and S. Desa, “Distributed MPC for dynamic supply chain management,” *Assessment and Future Directions of Nonlinear Model Predictive Control*, vol. 358, pp. 607–615, 2007.

[18] R. B. Myerson, *Game theory: analysis of conflict*. Harvard University Press, 1991.

[19] R. J. Aumann, “Game theory, the new palgrave,” *A Dictionary of Economics*, vol. 2, pp. 460–482, 1987.

[20] X. Du, Y. Xi, and S. Li, “Distributed model predictive control for large-scale systems,” *Proceedings of the 2001 American Control Conference*, pp. 3142–3143, 2001.

[21] S. Li, Y. Zhang, and Q. Zhu, “Nash-optimization enhanced distributed model predictive control applied to the Shell benchmark problem,” *Information Sciences*, vol. 170, no. 2-4, pp. 329–349, 2005.

[22] J. Von Neumann, O. Morgenstern, H. Kuhn, and A. Rubinstein, *Theory of games and economic behavior*. Princeton university press Princeton, NJ, 1947.

[23] L. Giovanini and J. Balderud, “Game approach to distributed model predictive control,” in *International Control Conference*.

[24] J. M. Muñoz, de la Peña, D. Maestre, and E. F. Camacho, “Distributed MPC Based on a Cooperative Game,” *48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009.

[25] P. A. Trodden, D. Nicholson, and A. G. Richards, “Distributed model predictive control as a game with coupled contraints,” *Proceedings of the European Control Conference*, 2009.

[26] J. F. Nash, “Equilibrium points in N-persons games,” *Proceedings of the National Academy of Sciences of United States of America*, vol. 36, no. 1, pp. 48–40, 1950.

[27] J. Nash, “The bargaining problem,” *Econometrica*, vol. 18, no. 2, pp. 155–162, 1950.

[28] J. Nash, “Two-person cooperative games,” *Econometrica*, vol. 21, no. 1, pp. 128–140, 1953.

[29] A. M. Brandenburger and B. J. Nalebuff, “The right game: Use the game theory to shape strategy,” *Harvadr Bussines Review*, July-August 1995.

[30] H. J. M. Peters, *Axiomatic Bargaining Game Theory*. Dordrecht: Kluwer Academic Publishers, 1992.

[31] C. Savorgnan and D. Moritz, "Control benchmark of a hydro-power plant (draft)," Tech. rep., Hierarchical and Distributed Model Predictive Control for Large-Scale Systems (HD-MPC), Katholieke Universiteit Leuven, March 23 2011.

[32] O. Akira, "A noncooperative approach to general n-person cooperative games," Discussion Papers 2005-01, Graduate School of Economics, Hitotsubashi University, 2005.

[33] J. C. Harsanyi, "A simplified bargaining model for the n-person cooperative game," *International Economic Review*, vol. 4, no. 2, pp. 194–220, 1963.

[34] J. Duggan and T. Kalandrakis, "A Newton Collocation Method for Solving Dynamic Bargaining Games," Wallis Working Papers WP60, University of Rochester - Wallis Institute of Political Economy, 2009.

[35] A. Haurie and B. Tolwinski, "Acceptable equilibria in dynamic bargaining games," in *System Modeling and Optimization* (R. Drenick and F. Kozin, eds.), vol. 38 of *Lecture Notes in Control and Information Sciences*, pp. 715–725, Springer Berlin / Heidelberg, 1982.

[36] A. Haurie and B. Tolwinski, "Acceptable equilibria in dynamic bargaining games," in *System Modeling and Optimization* (R. Drenick and F. Kozin, eds.), vol. 38 of *Lecture Notes in Control and Information Sciences*, pp. 715–725, Springer Berlin / Heidelberg, 1982.

[37] I. Alvarado, D. Limon, D. Munoz de la Pena, J.M. Maestre, R.R. Negenborn, F. Valencia, Holger Scheu, M.A. Ridao, B. De Schutter, J. Espinosa, and W. Marquardt, "A comparative analysis of distributed mpc techniques applied to the hd-mpc four-tank benchmark," *Journal of Process Control*, 2010.

[38] F. Valencia, J. Espinosa, B. De Schutter, and K. Staňková, "Feasible-cooperation distributed model predictive control scheme based on game theory," in *18th IFAC world congress*, 2010.

[39] A. Venkat, J. Rawlings, and S. Wright, "Distributed model predictive control of large-scale systems," in *Assessment and Future Directions of Nonlinear Model Predictive Control* (R. Findeisen, F. Allgower, and L. Biegler, eds.), vol. 358 of *Lecture Notes in Control and Information Sciences*, pp. 591–605, Springer Berlin / Heidelberg, 2007.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[41] S. Karlin, *Mathematical Methods and Theory in Games, Programming, and Economics Two Volumes Bound As One*. Dover Publ, 1992.

[42] R. Rockafellar, *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970.

[43] T. S. H. Driessen, "A survey of consistency properties in cooperative game theory," *SIAM Review*, 1991.

[44] Aswin N. Venkat, Ian A. Hiskens, James B. Rawlings, and S. J. Wright, "Distributed mpc strategies for automatic generation control," in *IFAC Symposium on Power Plants and Power Systems Control*, (Kananaskis, Canada), June 2006.

[45] K. H. Johansson, A. Horch, O. Wijk, K. Henrik, J. Alex, and A. Hansson, "Teaching multivariable control using the quadruple-tank process," 2000.

[46] C. Setz, A. Heinrich, P. Rostalski, G. Papafotiou, and M. Morari, “Application of model predictive control to a cascade of river power plants,” in *Proceedings of the 17th IFAC World Congress*, (Seul, Korea), July 6-11 2008.

[47] F. Petrone, *Model Predictive Control of a Hydro-Power Valley*. PhD thesis, Facolta di Ingegneria dell’Informazione Corso di Laurea Specialistica in Ingegneria dell’Atomazione, Politecnico di Milano, 2010.

[48] Gabriela Glanzmann, Martin von Siebenthal, Tobias Geyer, George Papafotiou, and M. Morari, “Supervisory water level control for cascaded river power plants,” Tech. rep., Swiss Federal Institute of Technology, Automatic Control Laboratory, CH-8092 Zurich, Switzerland, 2003.

Chapter 4

A Distributed Optimization-Based Approach for Hierarchical MPC of Large-Scale Systems with Coupled Dynamics and Constraints

The research of this chapter has been developed by Minh Dang Doan, Tamás Keviczky, and Bart De Schutter. The authors are with Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands

4.1 Introduction

Coordination and control of interacting subsystems is an essential requirement for optimal operation and enforcement of critical operational constraints in large-scale industrial processes and infrastructure systems [1]. Model Predictive Control (MPC) has become the method of choice when designing control systems for such applications [2, 3, 4], due to its ability to handle important process constraints explicitly. MPC relies on solving finite-time optimal control problems repeatedly online, which may become prohibitive for large-scale systems due to the problem size or communication constraints. Recent efforts have been focusing on how to decompose the underlying optimization problem in order to arrive at a distributed or hierarchical control system that can be implemented under the prescribed computational and communication limitations [5, 6]. One common way to decompose an MPC problem with coupled dynamics or constraints is to use dual decomposition methods [7, 8, 9], which typically lead to iterative algorithms (in either a distributed or hierarchical framework) that converge to feasible solutions only asymptotically. Implementing such approaches within each MPC update period can be problematic for some applications.

Recently, we have presented a dual decomposition scheme for solving large-scale MPC problems with coupling in both dynamics and constraints, where primal feasible solutions can be obtained even after a finite number of iterations [10]. In the current paper we present a novel method that is motivated by the use of constraint tightening in robust MPC [11], along with a primal averaging scheme and distributed Jacobi optimization. Since an exact optimum of the Lagrangian is not assumed to be computable in finitely many iterations, an approximate scheme is needed for solving the MPC optimization problem in each time step. We present a solution approach that requires a nested two-layer iteration structure and the sharing of a few crucial parameters in a hierarchical fashion. The proposed framework guarantees primal feasible solutions and MPC stability using a finite number of iterations with bounded suboptimality.

The paper is organized as follows. In Section 4.2, we describe the MPC optimization problem and its tightened version, which will be used to guarantee feasibility of the original problem even with a suboptimal primal solution. Section 4.3 describes the main elements of the algorithm used to solve the dual version of the tightened optimization problem: the approximate subgradient method and the distributed Jacobi updates. In Section 4.4, we show that the primal average solution generated by the approximate subgradient algorithm is a feasible solution of the original optimization problem, and that the cost function decreases through the MPC updates. This allows it to be used as a Lyapunov function for showing closed-loop MPC stability. Section 4.6 concludes the paper and outlines future research.

4.2 Problem description

4.2.1 MPC problem

We consider M interconnected subsystems with coupled discrete-time linear time-invariant dynamics:

$$x^i_{k+1} = \sum_{j=1}^{M} A^{ij} x^j_k + B^{ij} u^j_k, \quad i = 1,\dots,M \tag{4.1}$$

and the corresponding centralized state-space model:

$$x_{k+1} = A x_k + B u_k \tag{4.2}$$

with $x_k = [(x^1_k)^T (x^2_k)^T \dots (x^M_k)^T]^T$, $u_k = [(u^1_k)^T (u^2_k)^T \dots (u^M_k)^T]^T$, $A = [A_{ij}]_{i,j\in\{1,\dots,M\}}$ and $B = [B_{ij}]_{i,j\in\{1,\dots,M\}}$.

The MPC problem at time step t is formed using a convex cost function and convex constraints:

$$\min_{\mathbf{u},\mathbf{x}} \sum_{k=t}^{t+N-1} \left(x_k^T Q x_k + u_k^T R u_k \right) + x_{t+N}^T P x_{t+N} \tag{4.3}$$

$$\text{s.t.} \quad x^i_{k+1} = \sum_{j\in\mathcal{N}^i} A^{ij} x^j_k + B^{ij} u^j_k,$$

$$i = 1,\dots,M, \quad k = t,\dots,t+N-1 \tag{4.4}$$

$$x_k \in \mathscr{X}, k = t+1,\dots,t+N-1 \tag{4.5}$$

$$x_{t+N} \in \mathscr{X}_\mathrm{f} \subset \mathscr{X} \tag{4.6}$$

$$u_k \in \mathscr{U}, k = t,\dots,t+N-1 \tag{4.7}$$

$$u^i_k \in \Omega_i, i = 1,\dots,M, \quad k = t,\dots,t+N-1 \tag{4.8}$$

$$x_t = x(t) \in \mathscr{X} \tag{4.9}$$

where $\mathbf{u} = [u_t^T,\dots,u_{t+N-1}^T]^T$, $\mathbf{x} = [x_{t+1}^T,\dots,x_{t+N}^T]^T$, the matrices Q, P, and R are block-diagonal and positive definite, the constraint sets $\mathscr{U}$, $\mathscr{X}$ and $\mathscr{X}_\mathrm{f}$ are polytopes and have nonempty interiors, and each local constraint set Ω_i is a hyperbox. Each subsystem i is assigned a neighborhood, denoted $\mathcal{N}^i$, containing subsystems that have direct dynamical interactions with subsystem i, including itself. The initial state x_t is the current state at time step t.

As $\mathcal{U}$, $\mathcal{X}$ and $\mathcal{X}_\mathrm{f}$ are polytopes, the constraints (4.5) and (4.6) are represented by linear inequalities. Moreover, the state vector $\mathbf{x}$ is affinely dependent on $\mathbf{u}$. Hence, we can eliminate state variables $x_{t+1},\ldots,x_{t+N}$ and transform the constraints (4.4), (4.5), and (4.6) into linear inequalities of the input variable $\mathbf{u}$. Eliminating the state variables in (4.3)–(4.9) leads to an optimization problem in the following form:

$$f_t^* = \min_{\mathbf{u}} \quad f(\mathbf{u},x_t) \tag{4.10}$$

$$\text{s.t.} \quad g(\mathbf{u},x_t) \leq 0 \tag{4.11}$$

$$\mathbf{u} \in \mathbf{\Omega} \tag{4.12}$$

where f and $g = [g_1,\ldots,g_m]^T$ are convex functions, and $\mathbf{\Omega} = \prod_{i=1}^{M} \mathbf{\Omega}_i$ with each $\mathbf{\Omega}_i = \prod_{k=0}^{N-1} \Omega_i$ is a hyperbox. Note that $f(\mathbf{u},x_t) > 0, \forall \mathbf{u} \neq 0, x_t \neq 0$, due to the positive definiteness of Q, P, and R.

We will use $(\mathbf{u}_t,\mathbf{x}_t)$ to denote a feasible solution generated by the controller for problem (4.3)–(4.9) at time step t. This solution is required to be feasible but not necessarily optimal.We will make use of the following assumptions:

Assumption 4.2.1 *There exists a block-diagonal feedback gain K such that the matrix $A+BK$ is Schur (i.e., a decentralized stabilizing control law for the unconstrained aggregate system).*

Assumption 4.2.2 *The terminal constraint set $\mathcal{X}_\mathrm{f}$ is positively invariant for the closed-loop $x_{k+1} = (A+BK)x_k$ ($x \in \mathrm{int}(\mathcal{X}_\mathrm{f}) \Rightarrow (A+BK)x \in \mathrm{int}(\mathcal{X}_\mathrm{f})$).*

Assumption 4.2.3 *The Slater condition holds for problem* (4.10)–(4.12)*, i.e., there exists a vector that satisfies strict inequality constraints [12]. It is also assumed that prior to each time step t, a Slater vector $\bar{\boldsymbol{u}}_t$ is available, such that*

$$g_j(\bar{\boldsymbol{u}}_t,x_t) < 0, j = 1,\ldots,m \tag{4.13}$$

Remark 4.2.4 *Since $g(\boldsymbol{u},x_t) \leq 0$ has a nonempty interior, so do its components $g_j(\boldsymbol{u},x_t) \leq 0, j = 1,\ldots,m$. Hence, there will always be a vector that satisfies the Slater condition* (4.13)*. In fact, we will only need to find the Slater vector $\bar{\boldsymbol{u}}_0$ for the first time step, which can be computed off-line. In Section 4.5.1 we will show that a new Slater vector can then be obtained for each $t \geq 1$, using Assumption 4.2.2.*

Assumption 4.2.5 *At each time step t, the following holds*

$$f(\boldsymbol{u}_{t-1},x_{t-1}) - f(\bar{\boldsymbol{u}}_t,x_t) > x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} \tag{4.14}$$

For later reference, we define $\Delta_t > 0$ which can be computed before time step t as follows:

$$\Delta_t = x_{t-1}^T Q x_{t-1} + u_{t-1}^T R u_{t-1} \tag{4.15}$$

Remark 4.2.6 *Assumption 4.2.5 is often satisfied with an appropriate terminal penalty matrix P. A method to construct a block-diagonal P with a given decentralized stabilizing control law is provided in [13].*

Assumption 4.2.7 *For each $x_t \in \mathcal{X}$, the Euclidean norm of $g(\boldsymbol{u},x_t)$ is bounded:*

$$L_t \geq \|g(\boldsymbol{u},x_t)\|_2, \forall \boldsymbol{u} \in \mathbf{\Omega} \tag{4.16}$$

Remark 4.2.8 *In the first time step, with given x_0, we can find L_0 by evaluating $\|g(\boldsymbol{u},x_0)\|_2$ at the vertices of $\mathbf{\Omega}$, the maximum will then satisfy* (4.16) *for $t = 0$, due to the convexity of g and $\mathbf{\Omega}$. For the subsequent time steps, we will present a simple method to update L_t in Section 4.5.2.*

4.2.2 The tightened problem

We will not solve problem (4.10)–(4.12) directly. Instead, we will make use of an iterative algorithm based on a tightened version of (4.10)–(4.12). Consider the tightened constraint:

$$g'(\mathbf{u},x_t) \triangleq g(\mathbf{u},x_t) + \mathbf{1}_m c_t \leq 0 \tag{4.17}$$

with $g'(\mathbf{u},x_t) = [g'_1,\ldots,g'_m]^T$, $0 < c_t < \min_{j=1,\ldots,m}\{-g_j(\bar{\mathbf{u}}_t,x_t)\}$, and $\mathbf{1}_m$ the column vector with every entry equal to 1. Due to (4.13), we have

$$\min_{j=1,\ldots,m}\{g'_j(\bar{\mathbf{u}}_t,x_t)\} = \min_{j=1,\ldots,m}\{g_j(\bar{\mathbf{u}}_t,x_t)\} + c_t < 0 \tag{4.18}$$

Hence $g'_j(\bar{\mathbf{u}}_t,x_t) < 0, j = 1,\ldots,m$. Moreover, using (4.16) and the triangle inequality of the 2-norm, we will get $L'_t = L_t + c_t$ as the norm bound for g', i.e. $L'_t \geq \|g'(\mathbf{u},x_t)\|_2, \forall \mathbf{u} \in \mathbf{\Omega}$. Note that L'_t implicitly depends on x_t, as $\bar{\mathbf{u}}_t$ and c_t are updated based on the current state x_t.

Using the tightened constraint (4.17), we formulate the tightened problem:

$$f_t'^* = \min_{\mathbf{u}} \quad f(\mathbf{u},x_t) \tag{4.19}$$

$$\text{s.t.} \quad g'(\mathbf{u},x_t) \leq 0 \tag{4.20}$$

$$\mathbf{u} \in \mathbf{\Omega} \tag{4.21}$$

Remark 4.2.9 *Only the coupled constraints* (4.11) *are tightened, while the local input constraints* (4.12) *are unchanged. The Slater condition also holds for the tightened problem* (4.19)–(4.21)*, with* $\bar{\mathbf{u}}_t$ *being the Slater vector.*

4.3 The proposed optimization algorithm

Our objective is to calculate a feasible solution for problem (4.3)–(4.9) using a method that is favorable for distributed computation. The main idea is to use dual decomposition for the tightened problem (4.19)–(4.21) instead of the original one, such that after a finite number of iterations the constraint violations in the tightened problem will be less than the difference between the tightened and the original constraints. Thus, even after a finite number of iterations, we will obtain a primal feasible solution for the original MPC optimization problem.

4.3.1 The dual problem

We will tackle the dual problem of (4.19)–(4.21), in order to deal with coupled constraint $g'(\mathbf{u},x_t) \leq 0$ in a distributed way. In this section, we define the dual problem and its subgradient. For simplicity, in this section the dependence of functions on the initial condition x_t is not indicated explicitly.

The Lagrangian of problem (4.19)–(4.21) is defined as:

$$\mathscr{L}'(\mathbf{u},\mu) = f(\mathbf{u}) + \mu^T g'(\mathbf{u}) \tag{4.22}$$

in which $\mathbf{u} \in \mathbf{\Omega}, \mu \in \mathbb{R}^m_+$.

The dual function for (4.19)–(4.21):

$$q'(\mu) = \min_{\mathbf{u}\in\mathbf{\Omega}} \mathscr{L}'(\mathbf{u},\mu) \tag{4.23}$$

is a concave function on $\mathbb{R}^m_+$, and it is non-smooth when f and g' are not strictly convex functions [12].

Given the assumption that Slater condition holds for (4.19)–(4.21), duality theory [12] shows that:

$$q_t'^* = f_t'^* \tag{4.24}$$

with $q_t'^* = \max_{\mu\in\mathbb{R}^m_+} q'(\mu)$ and $f_t'^*$ the minimum of (4.19)–(4.21).

Thanks to this result, instead of minimizing the primal problem, we may maximize the dual problem, which is often more amenable to decomposition due to simpler constraints. Since we may not have the gradient of q' in all points of $\mathbb{R}^m_+$, we will use a method based on the subgradient.

Definition 4.3.1 *A vector d is called a subgradient of a convex function f over $\mathscr{X}$ at the point $x \in \mathscr{X}$ if:*

$$f(y) \geq f(x) + (y-x)^T d, \quad \forall y \in \mathscr{X} \tag{4.25}$$

The set of all subgradients of f at the point x is called the subdifferential of f at x, denoted $\partial f(x)$.

For each Lagrange multiplier $\bar{\mu} \in \mathbb{R}^m_+$, first assume we have $\mathbf{u}(\bar{\mu}) = \arg\min_{\mathbf{u}\in\mathbf{\Omega}} \mathscr{L}'(\mathbf{u},\bar{\mu})$. Then a subgradient of the dual function is directly available, since [12]:

$$q'(\mu) \leq q'(\bar{\mu}) + (\mu - \bar{\mu})^T g'(\mathbf{u}(\bar{\mu})), \forall \mu \in \mathbb{R}^m_+ \tag{4.26}$$

In case an optimum of the Lagrangian is not attained due to termination of the optimization algorithm after a finite number of steps, a value $\tilde{\mathbf{u}}(\bar{\mu})$ that satisfies

$$\mathscr{L}'(\tilde{\mathbf{u}}(\bar{\mu}),\bar{\mu}) \leq \min_{\mathbf{u}\in\mathbf{\Omega}} \mathscr{L}'(\mathbf{u},\bar{\mu}) + \delta \tag{4.27}$$

will lead to the following inequality:

$$q'(\mu) \leq q'(\bar{\mu}) + \delta + (\mu - \bar{\mu})^T g'(\tilde{\mathbf{u}}(\bar{\mu})), \forall \mu \in \mathbb{R}^m_+ \tag{4.28}$$

where $g'(\tilde{\mathbf{u}}(\bar{\mu}))$ is called a δ-subgradient of the dual function q at the point $\bar{\mu}$. The set of all δ-subgradients of q at $\bar{\mu}$ is called δ-subdifferential of q at $\bar{\mu}$.

This means we do not have to look for the subgradient (or δ-subgradient) of the dual function, it is available by just evaluating the constraint function at the primal value $\mathbf{u}(\bar{\mu})$ (or $\tilde{\mathbf{u}}(\bar{\mu})$).

4.3.2 The main algorithm

We organize our algorithm for solving (4.10)–(4.12) at time step t in a nested iteration of an outer and inner loop. The main procedure is described as follows:

Algorithm 4.3.2 ***Approximate subgradient method with nested Jacobi iterations***

1. *Given a Slater vector $\bar{\mathbf{u}}_t$ of* (4.10)–(4.12)*, determine c_t and construct the tightened problem* (4.19)–(4.21).
2. *Determine step size α_t and suboptimality ε_t, see later in Section 4.3.3.*
3. *Determine $\bar{k}_t$ (the sufficient number of outer iterations), see later in Section 4.3.3.*

4. ***Outer loop:*** *Set* $\mu^{(0)} = 0 \cdot \mathbf{1}_m$. *For* $k = 0, \ldots, \bar{k}$, *find* $\boldsymbol{u}^{(k)}, \mu^{(k+1)}$ *such that:*

$$\mathscr{L}'(\boldsymbol{u}^{(k)}, \mu^{(k)}) \leq \min_{\boldsymbol{u} \in \boldsymbol{\Omega}} \mathscr{L}'(\boldsymbol{u}, \mu^{(k)}) + \varepsilon_t \tag{4.29}$$

$$\mu^{(k+1)} = \mathscr{P}_{\mathbb{R}_+^m} \left\{ \mu^{(k)} + \alpha_t d^{(k)} \right\} \tag{4.30}$$

where $\mathscr{P}_{\mathbb{R}_+^m}$ *denotes the projection onto the nonnegative orthant,* $d^{(k)} = g'\left(\boldsymbol{u}^{(k)}, x_t\right)$.

Inner loop:

- *Determine* $\bar{p}_k$ *(the sufficient number of inner iterations), see later in Section 4.3.4.*
- *Solve problem* (4.29) *in a distributed way with a Jacobi algorithm. For* $p = 0, \ldots, \bar{p}_k$, *every subsystem* i *computes:*

$$\begin{aligned} \boldsymbol{u}^i(p+1) = \arg\min_{\boldsymbol{u}_i \in \boldsymbol{\Omega}_i} \mathscr{L}'(&\boldsymbol{u}_1(p), \ldots, \boldsymbol{u}_{i-1}(p), \boldsymbol{u}_i, \\ &\boldsymbol{u}_{i+1}(p), \ldots, \boldsymbol{u}_M(p), \mu^{(k)}) \end{aligned} \tag{4.31}$$

  where $\boldsymbol{\Omega}_i$ *is the local constraint set for control variables of subsystem* i.
- *Define* $\boldsymbol{u}^{(k)} \triangleq [\boldsymbol{u}^1(\bar{p}_k)^T, \ldots, \boldsymbol{u}^M(\bar{p}_k)^T]^T$, *which is guaranteed to satisfy* (4.29).

5. *Compute* $\hat{\mathbf{u}}^{(\bar{k}_t)} = \frac{1}{\bar{k}_t} \sum_{l=0}^{\bar{k}_t} \boldsymbol{u}^{(l)}$, *take* $\boldsymbol{u}_t = \hat{\boldsymbol{u}}^{(\bar{k}_t)}$ *as the solution of* (4.10)–(4.12).

Remark 4.3.3 *Algorithm 4.3.2 is suitable for implementation in a hierarchical fashion where the main computations occur in the Jacobi iterations and are executed by local controllers in parallel, while the updates of dual variables and common parameters are carried out by a higher-level coordinating controller. This algorithm is also amenable to implementation in distributed settings, where there are communication links available to help determine and propagate the common parameters* $\alpha_t, \varepsilon_t, \bar{k}_t$, *and* $\bar{p}_k$.

In the following sections, we will describe in detail how the computations are derived, and what the resulting properties are.

4.3.3 Outer loop: Approximate subgradient method

The outer loop at iteration k uses an approximate subgradient method. The primal average sequence $\hat{\mathbf{u}}^{(k)} = \frac{1}{k} \sum_{l=0}^{k} \mathbf{u}^{(l)}$ has the following properties:

$$\begin{aligned} &\text{For } k \geq 1: \\ &\left\| \left[g'\left(\hat{\mathbf{u}}^{(k)}, x_t\right) \right]^+ \right\|_2 \leq \frac{1}{k\alpha_t} \left(\frac{3}{\gamma_t} [f(\bar{\mathbf{u}}_t, x_t) - q_t'^*] \right. \\ &\qquad\qquad \left. + \frac{\alpha_t L_t'^2}{2\gamma_t} + \alpha_t L_t' \right) \end{aligned} \tag{4.32}$$

$$f\left(\hat{\mathbf{u}}^{(k)}, x_t\right) \leq f_t'^* + \frac{\left\|\mu^{(0)}\right\|_2^2}{2k\alpha_t} + \frac{\alpha_t L_t'^2}{2} + \varepsilon_t \tag{4.33}$$

where g'^+ denotes the constraint violation, i.e. $g'^+ = \max\{g', 0 \cdot \mathbf{1}_m\}$. The proof of (4.32) can be found in [14], and the proof of (4.33) is given in Appendix 4.7.1.

Determining α_t and ε_t

Using the lower bound of the cost reduction (4.14) and the upper bound of the suboptimality (4.33) for the tightened problem (4.19)–(4.21), we will choose α_t and ε_t such that $f(\mathbf{u}_t, x_t) < f(\mathbf{u}_{t-1}, x_{t-1})$.

The step size α_t and suboptimality ε_t should satisfy:

$$\frac{\alpha_t L_t'^2}{2} + \varepsilon_t \leq \Delta_t \tag{4.34}$$

where Δ_t is defined in (4.15), and L_t' is the norm bound for g'. This condition allows us to show the decreasing property of the cost function in problem (4.3)–(4.9), which can then be used as a Lyapunov function.

Note that a larger α_t will lead to a smaller number of outer iterations, while a larger ε_t will lead to a smaller number of inner iterations. For the remainder of the paper we choose their values according to

$$\alpha_t = \frac{\Delta_t}{L_t'^2} \tag{4.35}$$

$$\varepsilon_t = \frac{\Delta_t}{2} \tag{4.36}$$

Determining $\bar{k}_t$

Using the constraint violation bound (4.32), we will choose $\bar{k}_t$ such that at the end of the algorithm, we will get a feasible solution for problem (4.10)–(4.12), which is the average of primal iterates generated by (4.29):

$$\hat{\mathbf{u}}^{(\bar{k}_t)} = \frac{1}{\bar{k}_t} \sum_{l=0}^{\bar{k}_t} \mathbf{u}^{(l)} \tag{4.37}$$

The subgradient iteration (4.29)–(4.30) is performed for $k = 1, \ldots, \bar{k}_t$, with the integer

$$\bar{k}_t = \left\lceil \frac{1}{\alpha_t c_t} \left(\frac{3}{\gamma_t} f(\bar{\mathbf{u}}_t, x_t) + \frac{\alpha_t L_t'^2}{2\gamma_t} + \alpha_t L_t' \right) \right\rceil \tag{4.38}$$

defined *a priori*, where $\lceil \cdot \rceil$ is the ceiling operator which gives the closest integer equal to or above a real value, $\gamma_t = \min_{j=1,\ldots,m}\{-g_j'(\bar{\mathbf{u}}_t, x_t)\} = \min_{j=1,\ldots,m}\{-g_j(\bar{\mathbf{u}}_t, x_t)\} - c_t$, and $\bar{\mathbf{u}}_t$ is the Slater vector of (4.19)–(4.21).

4.3.4 Inner loop: Jacobi method

The inner iteration (4.31) performs parallel local optimizations based on a standard Jacobi distributed optimization method for a convex function $\mathscr{L}'(\mathbf{u}, \mu^{(k)})$ over a Cartesian product, as described in [15, Section 3.3]. In order to find the sufficient stopping condition of this Jacobi iteration, we need to characterize the convergence rate of this algorithm. In the following, we summarize the condition for convergence of the Jacobi iteration, noting that $\mathscr{L}'(\mathbf{u}, \mu^{(k)})$ is a convex quadratic function with respect to $\mathbf{u}$.

Proposition 4.3.4 *Suppose the following condition holds:*

$$\lambda_{\min}(H_{ii}) > \sum_{j\neq i} \bar{\sigma}(H_{ij}), \forall i \tag{4.39}$$

where H_{ij} with $i, j \in \{1,\dots,M\}$ denotes a submatrix of the Hessian H of $\mathscr{L}'$ w.r.t. $\boldsymbol{u}$, containing entries of H in rows belonging to subsystem i and columns belonging to subsystem j, $\lambda_{\min}$ means the smallest eigenvalue, and $\bar{\sigma}$ denotes the maximum singular value.

Then $\exists\phi \in (0,1)$ such that the aggregate solution of the Jacobi iteration (4.31) *satisfies:*

$$\|\boldsymbol{u}(p) - \boldsymbol{u}^*\|_2 \leq M\phi^p \max_i \|\boldsymbol{u}^i(0) - \boldsymbol{u}^{i*}\|_2, \quad \forall p \geq 1 \tag{4.40}$$

where $\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}\in\boldsymbol{\Omega}} \mathscr{L}'(\boldsymbol{u}, \mu^{(k)})$, and $\boldsymbol{u}^{i*}$ is the component of subsystem i in $\boldsymbol{u}^*$.

We provide a proof for Proposition 4.3.4 in Appendix 4.7.2.

Remark 4.3.5 *This proposition provides a linear convergence rate of the Jacobi iteration, under the condition of* weak dynamical couplings *between subsystems. For the sake of illustrating condition* (4.39)*, let all subsystems have the same number of inputs. Consequently, H_{ij} is a square and symmetric matrix for each pair (i, j), hence the maximum singular value $\bar{\sigma}(H_{ij})$ equals to the maximum eigenvalue. Inequality* (4.39) *thus reads:*

$$\lambda_{\min}(H_{ii}) > \sum_{j\neq i} \lambda_{\max}(H_{ij}), \forall i$$

which implies that the couplings represented by H are small in comparison with each local cost.

Remark 4.3.6 *Note that condition* (4.39) *is required only for the convergence rate result of the Jacobi iteration. Extensions to other types of systems, where the Lagrangian can be solved with bounded suboptimality, are immediate. In such cases we simply need to replace the Jacobi iteration with the new algorithm in the inner loop, while the outer loop will remain intact.*

Determining $\bar{p}_k$

As $\mathscr{L}'(\mathbf{u}, \cdot)$ is continuously differentiable in a closed bounded set $\boldsymbol{\Omega}$, it is Lipschitz continuous.

Suppose we know the Lipschitz constant Λ of $\mathscr{L}'(\mathbf{u}, \cdot)$ over $\boldsymbol{\Omega}$, i.e. for any $\mathbf{u}^1, \mathbf{u}^2 \in \boldsymbol{\Omega}$ the following inequality holds:

$$\|\mathscr{L}'(\mathbf{u}^1, \mu^{(k)}) - \mathscr{L}'(\mathbf{u}^2, \mu^{(k)})\|_2 \leq \Lambda \|\mathbf{u}^1 - \mathbf{u}^2\|_2 \tag{4.41}$$

Taking $\mathbf{u}^1 = \mathbf{u}(\bar{p}_k)$ and $\mathbf{u}^2 = \mathbf{u}^*$ in (4.41), and combining it with (4.40), we obtain:

$$\begin{aligned} \|\mathscr{L}'(\mathbf{u}(\bar{p}_k), \mu^{(k)}) - \min_{\mathbf{u}\in\boldsymbol{\Omega}} \mathscr{L}'(\mathbf{u}, \mu^{(k)})\|_2 &\leq \Lambda \|\mathbf{u}(\bar{p}_k) - \mathbf{u}^*\|_2 \\ &\leq \Lambda M \phi^{\bar{p}_k} \max_i \|\mathbf{u}^i(0) - \mathbf{u}^{i*}\|_2 \end{aligned} \tag{4.42}$$

For each $i \in \{1,\dots,M\}$, let D_i denote the diameter of the set $\boldsymbol{\Omega}_i$ w.r.t. the Euclidean norm, so we have $\|\mathbf{u}^i(0) - \mathbf{u}^{i*}\|_2 \leq D_i$. Hence the relation (4.42) can be further simplified as

$$\mathscr{L}'(\mathbf{u}(\bar{p}_k), \mu^{(k)}) \leq \min_{\mathbf{u}\in\boldsymbol{\Omega}} \mathscr{L}'(\mathbf{u}, \mu^{(k)}) + \Lambda M \phi^{\bar{p}_k} \max_i D_i \tag{4.43}$$

Based on (4.43), in order to use $\mathbf{u}(\bar{p}_k)$ as the solution $\mathbf{u}^{(k)}$ that satisfies (4.29), we choose the smallest integer $\bar{p}_k$ such that $\Lambda M \phi^{\bar{p}_k} \max_i D_i \leq \varepsilon_t$:

$$\bar{p}_k = \left\lceil \log_\phi \frac{\varepsilon_t}{\Lambda M \max_i D_i} \right\rceil \tag{4.44}$$

4.4 Properties of the algorithm

4.4.1 Distributed Jacobi algorithm with guaranteed convergence

The computations in the inner loop can be executed by subsystems in parallel. Let us define an r-step extended neighborhood of a subsystem i, denoted by $\mathcal{N}_r^i$, as the set containing all subsystems that can influence subsystem i within r successive time steps. $\mathcal{N}_r^i$ is the union of subsystem indices in the neighborhoods of all subsystems in $\mathcal{N}_{r-1}^i$:

$$\mathcal{N}_r^i = \bigcup_{j \in \mathcal{N}_{r-1}^i} \mathcal{N}^j \tag{4.45}$$

where $\mathcal{N}_1^i = \mathcal{N}^i$. We can see that in order to get update information in the Jacobi iterations, each subsystem i needs to communicate only with subsystems in $\mathcal{N}_{N-1}^i$, where N is the prediction horizon. This set includes all other subsystems that couple with i in the problem (4.10)–(4.12) after eliminating the state variables. This communication requirement indicates that we will benefit from communication reduction when the number of subsystems M is much larger than the horizon N, and the coupling structure is sparse.

Assume that the weak coupling condition (4.39) holds, then after $\bar{p}_k$ iterations as computed by (4.44), the Jacobi algorithm generates a solution $\mathbf{u}^{(k)} \triangleq \mathbf{u}(\bar{p}_k)$ that satisfies (4.29) in the outer loop.

4.4.2 Feasible primal solution

Proposition 4.4.1 *Suppose Assumptions 4.2.1 and 4.2.3 hold. Construct g' as in* (4.17)*, α_t as in* (4.35)*. Let the outer loop* (4.29)–(4.30) *with $\mu^{(0)} = 0 \cdot \mathbf{1}_m$ be iterated for $k = 0, \ldots, \bar{k}_t$. Then $\hat{\mathbf{u}}^{(\bar{k}_t)}$ is a feasible solution of* (4.10)–(4.12)*, where $\hat{\mathbf{u}}^{(\bar{k}_t)}$ is the primal average, computed by* (4.37).

Proof: With a finite number of $\bar{k}_t$ iterations (4.32) reads as

$$\left\| \left[g'\left(\hat{\mathbf{u}}^{(\bar{k}_t)}, x_t\right) \right]^+ \right\|_2 \leq \frac{1}{k_t \alpha_t} \left(\frac{3}{\gamma_t} \left[f(\bar{\mathbf{u}}_t, x_t) - q_t'^* \right] + \frac{\alpha_t L_t'^2}{2\gamma_t} + \alpha_t L_t' \right) \tag{4.46}$$

Moreover, the dual function q_t' is a concave function, therefore $q_t'^* \geq q'(0, x_t)$. Recall that $f(\mathbf{u}, x_t) > 0, \forall \mathbf{u} \neq 0, x_t \neq 0$, thus $q'(0, x_t) = \min_{\mathbf{u} \in \mathbf{\Omega}} f(\mathbf{u}, x_t) + 0 \cdot \mathbf{1}_m^T g'(\mathbf{u}, x_t) = \min_{\mathbf{u} \in \mathbf{\Omega}} f(\mathbf{u}, x_t) > 0$, thus

$$\left\| \left[g'\left(\hat{\mathbf{u}}^{(\bar{k}_t)}, x_t\right) \right]^+ \right\|_2 < \frac{1}{k_t \alpha_t} \left(\frac{3}{\gamma_t} f(\bar{\mathbf{u}}_t, x_t) + \frac{\alpha_t L_t'^2}{2\gamma_t} + \alpha_t L_t' \right) \tag{4.47}$$

Combining (4.47) with (4.38), and noticing that $\bar{k}_t$ and c_t are all positive lead to

$$\left\| \left[g'\left(\hat{\mathbf{u}}^{(\bar{k}_t)}, x_t\right) \right]^+ \right\|_2 < c_t \tag{4.48}$$

$$\Rightarrow g_j'\left(\hat{\mathbf{u}}^{(\bar{k}_t)}, x_t\right) < c_t, \quad j = 1, \ldots, m \tag{4.49}$$

$$\Rightarrow g_j\left(\hat{\mathbf{u}}^{(\bar{k}_t)}, x_t\right) < 0, \quad j = 1, \ldots, m \tag{4.50}$$

where the last inequality implies that $\hat{\mathbf{u}}^{(\bar{k}_t)}$ is a feasible solution of problem (4.10)–(4.12), due to $c_t < \min_{j=1,\ldots,m}\{-g_j(\bar{\mathbf{u}}_t, x_t)\}$. □

4.4.3 Closed-loop stability

Proposition 4.4.2 *Suppose Assumptions 4.2.3, 4.2.5, and 4.2.7 hold. Then the solution* $\hat{\mathbf{u}}^{(\bar{k}_t)}$ *generated by Algorithm 4.3.2 satisfies the following inequality:*

$$f(\boldsymbol{u}_t, x_t) < f(\boldsymbol{u}_{t-1}, x_{t-1}), \quad \forall t \in \mathbb{Z}_+ \tag{4.51}$$

Proof: Using (4.33) and (4.34), and noting that $\mu^{(0)} = 0$, we obtain:

$$f\left(\hat{\mathbf{u}}^{(\bar{k}_t)}, x_t\right) \leq f_t'^* + \frac{\|\mu^{(0)}\|}{2k_t\alpha_t} + \frac{\alpha_t L_t'^2}{2} + \varepsilon_t \leq f_t'^* + \Delta_t \tag{4.52}$$

Notice that $\bar{\mathbf{u}}_t$ is also a feasible solution of (4.19)–(4.21) (due to the way we construct the tightened problem: $\bar{\mathbf{u}}_t$ still belongs to the interior of the tightened constraint set), while $f_t'^*$ is the optimal cost value of this problem. As a consequence,

$$f_t'^* \leq f(\bar{\mathbf{u}}_t, x_t) \tag{4.53}$$

Combining (4.52), (4.53), and (4.14), and noting that $\mathbf{u}_t = \hat{\mathbf{u}}^{(\bar{k}_t)}$ leads to:

$$f(\mathbf{u}_t, x_t) < f(\mathbf{u}_{t-1}, x_{t-1}), \quad \forall t \in \mathbb{Z}_+ \tag{4.54}$$

□

Note that besides the decreasing property of $f(\mathbf{u}_t, x_t)$, all the other conditions for Lyapunov stability of MPC [16] are satisfied. Therefore, Proposition 4.4.2 leads to closed-loop MPC stability, where the cost function $f(\mathbf{u}_t, x_t)$ is a Lyapunov candidate function.

4.5 Realization of the assumptions

In this section, we discuss the way to update the Slater vector and the constraint norm bound for each time step, implying that Assumptions 4.2.3 and 4.2.7 are only necessary in the first time step ($t = 0$).

4.5.1 Updating the Slater vector

Lemma 4.5.1 *Suppose Assumption 4.2.2 holds. Let* $\boldsymbol{u}_t$ *be the solution of the MPC problem* (4.3)–(4.9) *at time step* t*, computed by Algorithm 4.3.2. Then* $\tilde{\mathbf{u}}_{t+1}$ *constructed by shifting* $\boldsymbol{u}_t$ *one step ahead and adding* $\tilde{u}_{t+N} = Kx_{t+N}$*, is a Slater vector for constraint* (4.11) *at time step* $t+1$.

Proof: Note that based on Proposition 4.4.1, $\hat{\mathbf{u}}^{(\bar{k}_t)}$ is a feasible solution of problem (4.10)–(4.12). Moreover, the strict inequality (4.50) means that $\hat{\mathbf{u}}^{(\bar{k}_t)}$ is in the interior of the constraint set of (4.3)–(4.9). This also yields:

$$x_{t+N} \in \text{int}(\mathscr{X}_\text{f}) \tag{4.55}$$

Moreover, due to Assumption 4.2.2, we have $(A+BK)x_{t+N} \in \text{int}(\mathscr{X}_\text{f})$. This means that if we use $\tilde{u}_{t+N} = Kx_{t+N}$, then the next state is also in the interior of the terminal constraint set $\mathscr{X}_\text{f}$. Note that $\mathscr{U}$ and $\mathscr{X}$ do not change when problem (4.3)–(4.9) is shifted from t to $t+1$, hence all the inputs of $\tilde{\mathbf{u}}_{t+1}$ and their subsequent states are in the interior of the corresponding constraint sets. Therefore, $\tilde{\mathbf{u}}_{t+1}$ as constructed at step 5 of Algorithm 4.3.2 is a Slater vector for the constraint (4.11) at time step $t+1$.□

This means we can use $\bar{\mathbf{u}}_{t+1} = \tilde{\mathbf{u}}_{t+1}$ as the qualifying Slater vector for Assumption 4.2.3 at time step $t+1$.

4.5.2 Updating the constraint norm bound

In our general problem setup, $g(\mathbf{u},x)$ is composed of affine functions over $\mathbf{u}$ and x, and thus can be written compactly as

$$g(\mathbf{u},x) = \Xi x + \Theta\mathbf{u} + \tau \tag{4.56}$$

with constant matrices Ξ,Θ and vector τ. Then for each x_{t-1}, x_t, and $\mathbf{u}\in\mathbf{\Omega}$, the following holds:

$$\begin{aligned} g(\mathbf{u},x_t) &= g(\mathbf{u},x_{t-1}) + \Xi(x_t - x_{t-1}) \\ \Rightarrow \|g(\mathbf{u},x_t)\|_2 &\leq \|g(\mathbf{u},x_{t-1})\|_2 + \|\Xi(x_t - x_{t-1})\|_2 \end{aligned} \tag{4.57}$$

In order to find a bound L_t for $g(\mathbf{u},x_t)$ in each $t\geq 1$ step, we assume to have the constraint norm bound available from the previous step:

$$L_{t-1} \geq \|g(\mathbf{u},x_{t-1})\|_2, \forall \mathbf{u}\in\mathbf{\Omega} \tag{4.58}$$

Hence, combining the above inequalities a norm bound update for $g(\mathbf{u},x_t)$ can be obtained as:

$$L_t = L_{t-1} + \|\Xi(x_t - x_{t-1})\|_2 \tag{4.59}$$

4.6 Conclusions

We have presented a constraint tightening approach for solving an MPC optimization problem with guaranteed feasibility and stability after a finite number of iterations. The new method is applicable to large-scale systems with coupling in dynamics and constraints, and the solution is based on approximate subgradient and Jacobi iterative methods, which facilitate implementation in a hierarchical or distributed way. Future extensions of this scheme include *a posteriori* choice of the solution by comparing the cost functions associated with the Slater vector $\bar{\mathbf{u}}_t$ and the primal average $\hat{\mathbf{u}}^{(k_t)}$ in a distributed way.

4.7 Appendix

4.7.1 Proof of the upper bound on the cost function (4.33)

This proof is an extension of the proof of Proposition 3(b) in [14], the main difference being the incorporation of the suboptimality ε_t in the update of the primal variable (4.29).

Using the convexity of the cost function, we have:

$$\begin{aligned} f(\hat{\mathbf{u}}^{(k)}) = f\left(\frac{1}{k}\sum_{l=0}^{k-1}\mathbf{u}^{(l)}\right) &\leq \frac{1}{k}\sum_{l=0}^{k-1} f(\mathbf{u}^{(l)}) \\ = \frac{1}{k}\sum_{l=0}^{k-1}\left(f(\mathbf{u}^{(l)}) + (\mu^{(l)})^T g'(\mathbf{u}^{(l)})\right) &- \frac{1}{k}\sum_{l=0}^{k-1}(\mu^{(l)})^T g'(\mathbf{u}^{(l)}) \end{aligned} \tag{4.60}$$

Note that $\mathscr{L}'\left(\mathbf{u}^{(l)},\mu^{(l)}\right) = \left(f(\mathbf{u}^{(l)}) + g'(\mathbf{u}^{(l)})^T\mu^{(l)}\right)$ and

$$\mathscr{L}'\left(\mathbf{u}^{(l)},\mu^{(l)}\right) \leq \min_{\mathbf{u}\in\mathbf{\Omega}} \mathscr{L}'\left(\mathbf{u}^{(l)},\mu^{(l)}\right) + \varepsilon_t = q'\left(\mu^{(l)}\right) + \varepsilon_t,$$

$$\forall l < k \tag{4.61}$$

Combining the two inequalities above, we then have:

$$\begin{aligned} f(\hat{\mathbf{u}}^{(k)}) &\leq \frac{1}{k}\sum_{l=0}^{k-1} q'\big(\mu^{(l)}\big) + \varepsilon_t - \frac{1}{k}\sum_{l=0}^{k-1}(\mu^{(l)})^T g'(\mathbf{u}^{(l)}) \\ &\leq q_t'^{*} + \varepsilon_t - \frac{1}{k}\sum_{l=0}^{k-1}(\mu^{(l)})^T d^{(l)} \end{aligned} \tag{4.62}$$

where $d^{(l)} = g'(\mathbf{u}^{(l)})$, and the last inequality is due to $q_t'^{*} \geq q'\big(\mu^{(l)}\big), \forall l$.

Using the expression of squared sum:

$$\begin{aligned} \|\mu^{(l+1)}\|_2^2 &\leq \|\mu^{(l)} + \alpha_t d^{(l)}\|_2^2 \\ &= \|\mu^{(l)}\|_2^2 + 2\alpha_t(\mu^{(l)})^T d^{(l)} + \|\alpha_t d^{(l)}\|_2^2 \end{aligned} \tag{4.63}$$

we have:

$$-(\mu^{(l)})^T d^{(l)} \leq \frac{1}{2\alpha_t}\left(\|\mu^{(l)}\|_2^2 - \|\mu^{(l+1)}\|_2^2 + \alpha_t^2\|d^{(l)}\|_2^2\right) \tag{4.64}$$

for $l = 0,\ldots,k-1$.

Summing side by side for $l = 0,\ldots,k-1$, we get:

$$\begin{aligned} -\sum_{l=0}^{k-1}(\mu^{(l)})^T d^{(l)} \leq & \frac{1}{2\alpha_t}\left(\|\mu^{(0)}\|_2^2 - \|\mu^{(k)}\|_2^2\right) \\ & + \frac{\alpha_t}{2}\sum_{l=0}^{k-1}\|d^{(l)}\|_2^2 \end{aligned} \tag{4.65}$$

Linking (4.62) and (4.65), we then have:

$$\begin{aligned} f(\hat{\mathbf{u}}^{(k)}) &\leq q_t'^{*} + \varepsilon_t + \frac{1}{2k\alpha_t}\left(\|\mu^{(0)}\|^2 - \|\mu^{(k)}\|^2\right) \\ &\quad + \frac{\alpha_t}{2k}\sum_{l=0}^{k-1}\|d^{(l)}\|^2 \\ &\leq q_t'^{*} + \frac{\|\mu^{(0)}\|^2}{2k\alpha_t} + \frac{\alpha_t L_t'^2}{2} + \varepsilon_t \end{aligned} \tag{4.66}$$

in which we get the last inequality by using L_t' as the norm bound for all $g'(\mathbf{u}^{(l)}), l = 0,\ldots,k-1$.

Finally, with the Slater condition, there is no primal-dual gap, i.e. $q_t'^{*} = f_t^{*}$ (cf. (4.24)), hence:

$$f(\hat{\mathbf{u}}^{(k)}) \leq f_t'^{*} + \frac{\|\mu^{(0)}\|^2}{2k\alpha_t} + \frac{\alpha_t L_t'^2}{2} + \varepsilon_t$$

□

4.7.2 Proof of the convergence result of the Jacobi iteration (Proposition 4.3.4)

According to Proposition 3.10 in [15, Chapter 3], the Jacobi algorithm has a linear convergence w.r.t. the block-maximum norm, as defined below:

Definition 4.7.1 *For each vector $x = [x_1^T, \ldots, x_M^T]$ with $x_i \in \mathbb{R}^{n_i}$, given a norm $\|\cdot\|_i$ for each i, the block-maximum norm based on $\|\cdot\|_i$ is defined as:*

$$\|x\|_{b\text{-}m} = \max_i \|x_i\|_i \tag{4.67}$$

Definition 4.7.2 *With any matrix $A \in \mathbb{R}^{n_i \times n_j}$, we associate the induced matrix norm of the block-maximum norm:*

$$\|A\|_{ij} = \max_{x \neq 0} \frac{\|Ax\|_i}{\|x\|_j} = \max_{\|x\|_j = 1} \|Ax\|_i \tag{4.68}$$

In this paper, we use the Euclidean norm as the default basis for block-maximum norm, i.e. $\|\cdot\|_i = \|\cdot\|_2, \forall i$.

Proposition 3.10 in [15, Chapter 3] states that $\mathbf{u}(p)$ generated by (4.31) will converge to the optimizer of $\mathscr{L}'(\mathbf{u}, x_t)$ with linear convergence rate w.r.t. block-maximum norm (i.e. $\|\mathbf{u}(p) - \mathbf{u}^*\|_{\text{b-m}} \leq \phi^p \|\mathbf{u}(0) - \mathbf{u}^*\|_{\text{b-m}}$, with $\mathbf{u}^* = \arg\min_{\mathbf{u}} \mathscr{L}'(\mathbf{u}, x_t)$ and $\phi \in [0,1)$) if there exists a positive scalar γ such that the mapping $R : \mathbf{\Omega} \mapsto \mathbb{R}^{n_\mathbf{u}}$, defined by $R(\mathbf{u}) = \mathbf{u} - \gamma \nabla_\mathbf{u} \mathscr{L}'(\mathbf{u}, x_t)$, is a contraction w.r.t. the block-maximum norm.

Our focus now is to derive the condition such that $R(\mathbf{u})$ is a contraction mapping.

Note that since $f(\mathbf{u}, x_t)$ is a quadratic function, and $g'(\mathbf{u}, x_t)$ contains only linear functions, the function $\mathscr{L}'(\mathbf{u}, x_t)$ is also a quadratic function w.r.t. $\mathbf{u}$, hence it can be written as:

$$\mathscr{L}'(\mathbf{u}, x_t) = \mathbf{u}^T H \mathbf{u} + b^T \mathbf{u} + c \tag{4.69}$$

where H is a symmetric, positive definite matrix, b is a constant vector and c is a constant scalar.

In order to derive the condition for $R(\mathbf{u})$ to be a contraction mapping, we will make use of Proposition 1.10 in [15, Chapter 3], stating that:

If $f : \mathbb{R}^{n_\mathbf{u}} \mapsto \mathbb{R}^{n_\mathbf{u}}$ is continuously differentiable and there exists a scalar $\phi \in [0,1)$ such that

$$\|I - \gamma G_i^{-1} \left(\nabla_i F_i(\mathbf{u})\right)^T\|_{ii} + \sum_{j \neq i} \|\gamma G_i^{-1} \left(\nabla_j F_i(\mathbf{u})\right)^T\|_{ij} \leq \phi, \quad \forall \mathbf{u} \in \mathbf{\Omega}, \forall i \tag{4.70}$$

then the mapping $T : \mathbf{\Omega} \mapsto \mathbb{R}^{n_\mathbf{u}}$ defined with each component $i \in \{1, \ldots, M\}$ by $T_i(\mathbf{u}) = \mathbf{u}_i - \gamma G_i^{-1} F(\mathbf{u})$ is a contraction with respect to the block-maximum norm.

The mapping $T(\mathbf{u})$ will become the mapping $R(\mathbf{u})$ if we take $G_i = I^{n_{\mathbf{u}^i}}, \forall i$ and $F(\mathbf{u}) = \nabla_\mathbf{u} \mathscr{L}'(\mathbf{u}, x_t) = 2H\mathbf{u} + b$. With such choice, and evaluating the induced matrix norm (4.68) in (4.70), the condition for contraction mapping of $R(\mathbf{u})$ is to find $\phi \in [0,1)$ such that:

$$\|I^{n_{\mathbf{u}^i}} - 2\gamma H_{ii}\|_2 + \sum_{j \neq i} \|2\gamma H_{ij}\|_2 \leq \phi, \forall i \tag{4.71}$$

where H_{ij} with $i, j \in \{1, \ldots, M\}$ denotes the submatrix of H, containing entries at rows belonging to subsystem i and columns belonging to subsystem j. Note that the matrix inside the first induced matrix norm is a square, symmetric matrix, while the matrices H_{ij} are generally not symmetric, depending on the number of variables of each subsystem. The scalar $\phi \in [0,1)$ is also the modulus of the contraction.

Using the properties of eigenvalue and singular value of matrices, we transform (4.71) into the following inequality:

$$\max_{\lambda} |2\gamma\lambda(H_{ii}) - 1| + 2\gamma \sum_{j \neq i} \bar{\sigma}(H_{ij}) \leq \phi, \forall i \tag{4.72}$$

where λ means eigenvalue, and $\bar{\sigma}$ denotes the maximum singular value.

In order to find $\gamma > 0$ and $\phi \in [0,1)$ satisfying (4.72), we need:

$$\max_{\lambda} |2\gamma\lambda(H_{ii}) - 1| + 2\gamma \sum_{j \neq i} \bar{\sigma}(H_{ij}) < 1, \forall i \tag{4.73}$$

$$\Leftrightarrow \left\{ \begin{array}{l} 2\gamma\lambda_{\max}(H_{ii}) - 1 + 2\gamma \sum_{j \neq i} \bar{\sigma}(H_{ij}) < 1 \\ 1 - 2\gamma\lambda_{\min}(H_{ii}) + 2\gamma \sum_{j \neq i} \bar{\sigma}(H_{ij}) < 1 \end{array} \right. , \forall i \tag{4.74}$$

$$\Leftrightarrow \left\{ \begin{array}{l} \gamma < 1/\left(\lambda_{\max}(H_{ii}) + \sum_{j \neq i} \bar{\sigma}(H_{ij})\right) \\ \lambda_{\min}(H_{ii}) > \sum_{j \neq i} \bar{\sigma}(H_{ij}) \end{array} \right. , \forall i \tag{4.75}$$

The first inequality of (4.75) shows how to choose γ, while the second inequality of (4.75) needs to be satisfied by the problem structure, which implies there are *weak dynamical couplings* between subsystems.

In summary, the mapping $R(\mathbf{u})$ satisfies (4.70) and thus is a contraction mapping if the following conditions hold:

1. For all i:

$$\lambda_{\min}(H_{ii}) > \sum_{j \neq i} \bar{\sigma}(H_{ij}) \tag{4.76}$$

2. The coefficient γ is chosen such that:

$$\gamma < \frac{1}{\lambda_{\max}(H_{ii}) + \sum_{j \neq i} \bar{\sigma}(H_{ij})}, \forall i \tag{4.77}$$

So, when condition (4.76) is satisfied and with γ chosen by (4.77), we can define $\phi \in (0,1)$ as:

$$\begin{aligned} \phi = \max_{i} \Bigg\{ \max \Bigg\{ & 2\gamma\big(\lambda_{\max}(H_{ii}) + \sum_{j \neq i} \bar{\sigma}(H_{ij})\big) - 1, \\ & 1 - 2\gamma\big(\lambda_{\min}(H_{ii}) - \sum_{j \neq i} \bar{\sigma}(H_{ij})\big) \Bigg\} \Bigg\} \end{aligned} \tag{4.78}$$

This ϕ is the modulus of the contraction $R(\mathbf{u})$, and also acts as the coefficient of the linear convergence rate of the Jacobi iteration (4.31), which means:

$$\|\mathbf{u}(p) - \mathbf{u}^*\|_{\text{b-m}} \leq \phi^p \|\mathbf{u}(0) - \mathbf{u}^*\|_{\text{b-m}}, \quad \forall p \geq 1 \tag{4.79}$$

where $\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathbf{\Omega}} \mathscr{L}'(\mathbf{u}, x_t)$.

Note that the closer of ϕ to 0, the faster the aggregate update $\mathbf{u}(p)$ converges to the optimizer of the Lagrange function.

In order to get the convergence rate w.r.t. the Euclidean norm, we will need to link from the Euclidean norm to the block-maximum norm:

$$\|x\|_2 \leq \sum_{i=1}^{M} \|x^i\|_2 \leq M \max_i \|x^i\|_2 = M\|x\|_{\text{b-m}} \tag{4.80}$$

Hence, the convergence rate of Jacobi iteration (4.31) w.r.t. the Euclidean norm is:

$$\|\mathbf{u}(p) - \mathbf{u}^*\|_2 \leq M\phi^p \max_i \|\mathbf{u}^i(0) - \mathbf{u}^{i*}\|_2, \quad \forall p \geq 1 \tag{4.81}$$

□

Bibliography

[1] J. B. Rawlings and B. T. Stewart, “Coordinating multiple optimization-based controllers: New opportunities and challenges,” *Journal of Process Control*, vol. 18, pp. 839–845, Oct. 2008.

[2] J. M. Maciejowski, *Predictive Control with Constraints*. Harlow, England: Prentice-Hall, 2002.

[3] E. F. Camacho and C. Bordons, *Model Predictive Control*. London: Springer, 1999.

[4] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill Publishing, 2009.

[5] R. Scattolini, “Architectures for distributed and hierarchical model predictive control - A review,” *Journal of Process Control*, vol. 19, pp. 723–731, May 2009.

[6] A. Venkat, I. Hiskens, J. Rawlings, and S. Wright, “Distributed MPC strategies with application to power system automatic generation control,” *IEEE Transactions on Control Systems Technology*, vol. 16, pp. 1192–1206, Nov. 2008.

[7] Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi, “Decentralized model predictive control via dual decomposition,” in *47th IEEE Conference on Decision and Control*, pp. 381–386, 2008.

[8] I. Necoara and J. Suykens, “Application of a smoothing technique to decomposition in convex optimization,” *IEEE Transactions on Automatic Control*, vol. 53, pp. 2674–2679, Dec. 2008.

[9] D. Doan, T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter, “A distributed version of Han’s method for DMPC using local communications only,” *Control Engineering and Applied Informatics*, vol. 11, pp. 6–15, Sept. 2009.

[10] M. D. Doan, T. Keviczky, and B. De Schutter, “A dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical MPC problems,” in *18th IFAC World Congress*, (Milan, Italy), Aug. 2011.

[11] Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How, “Distributed robust receding horizon control for multivehicle guidance,” *IEEE Transactions on Control Systems Technology*, vol. 15, 2007.

[12] D. P. Bertsekas, *Nonlinear programming*. Belmont, MA: Athena Scientific, 1999.

[13] D. D. Šiljak, *Large-scale dynamic systems: Stability and structure*. New York, NY: North Holland, 1978.

[14] A. Nedic and A. Ozdaglar, “Approximate primal solutions and rate analysis for dual subgradient methods,” *SIAM Journal on Optimization*, vol. 19, pp. 1757–1780, Nov. 2009.

[15] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ: Prentice-Hall, 1989.

[16] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, June 2000.