Delft University of Technology

Delft Center for Systems and Control

Technical report 15-024

A distributed algorithm to determine lower and upper bounds in branch and bound for hybrid model predictive control*

A. Firooznia, R. Bourdais, and B. De Schutter

If you want to cite this report, please use the following reference instead:
A. Firooznia, R. Bourdais, and B. De Schutter, "A distributed algorithm to determine lower and upper bounds in branch and bound for hybrid model predictive control," *Proceedings of the 54th IEEE Conference on Decision and Control*, Osaka, Japan, pp. 1736–1741, Dec. 2015.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

A Distributed Algorithm to Determine Lower and Upper Bounds in Branch and Bound for Hybrid Model Predictive Control

Amir Firooznia[1], Romain Bourdais[2], and Bart De Schutter[1]

***Abstract*— In this work, a class of model predictive control problems with mixed real-valued and binary control signals is considered. The optimization problem to be solved is a constrained Mixed Integer Quadratic Programming (MIQP) problem. The main objective is to derive a distributed algorithm for limiting the search space in branch and bound approaches by tightening the lower and upper bounds of objective function. To this aim, a distributed algorithm is proposed for the convex relaxation of the MIQP problem via dual decomposition. The effectiveness of the approach is illustrated with a case study.**

I. Introduction

For simple linear setups, the Model Predictive Control (MPC) problem can be formulated in the form of a Quadratic Programming (QP) problem . However, in case of hybrid systems [1], binary variables are introduced, which implies that the convex QP problem has to be replaced by a non-convex Mixed Integer Quadratic Programming (MIQP) optimization problem. In general, MIQP problems are NP-hard [2].

A commonly used approach for solving MIQP problems is branch and bound [3]. In this method, the subproblems are ordered in a tree structure, where at each level one new integer variable is fixed and a relaxed subproblem is solved. Depending on the problem, a large number of subproblems have to be solved and the worst case complexity is known to be exponential. The efficiency of the branch and bound method mainly depends on the possibility to efficiently compute tight bounds on the optimal objective function value during the branching, which reduces the search space. Commonly QP relaxations, in which integer constraints are relaxed to interval constraints, are applied to generate the subproblems that have to be solved in the nodes to produce the bounds in the branching procedure. However, it has been shown that is possible to use Semi-Definite Programming (SDP) to compute tighter bounds for integer programming problems [4]. SDP relaxations have previously been proposed for control of systems with binary inputs in [5], [6]. In [4], [7], different relaxations applicable to a hybrid MPC problem with binary control signals are compared. The result is that the standard SDP relaxation usually gives the best bound and is the most computationally demanding, while the QP relaxation gives the worst bound and is the least computationally demanding. The equality constrained relaxation presented in [4] often gives a better bound than the QP relaxation and is much less computationally demanding compared to the standard SDP relaxation. Furthermore, any small improvements in the bounds can reduce the size of branch and bound tree significantly.

Due to possible large number of the involved interconnected subsystems, which can result in huge number of variables and the inherent complexity, combinatorial explosion makes the MPC optimization problem for these systems difficult to solve and even sometimes infeasible for online implementation. The recent trend of research is to propose distributed algorithms for optimization in which the subsystems solve reduced optimization problems and interact with each other to obtain a solution close enough to the global value. To this purpose, communication between subsystems is required, and a coordination mechanism has to be developed. Different techniques have been developed to solve a global optimization problem using a decomposed scheme [8]–[16]. However, efficient distributed schemes in the hybrid systems context are still an open challenge [17]. In [18], a distributed method based on price decomposition is presented. As the convexity property is lost by the binary nature of the decision variables, a duality gap occurs while applying the dual decomposition [19]; so, a recursive procedure to find the optimal solution is proposed, which requires many iterations.

In this paper, a class of interconnected systems governed by real and binary inputs is considered in which the subsystems are linked to each other. This can be seen as a particular case of hybrid systems. Applications of such systems include finance, manufacturing systems, network and transportation problems, electricity networks, and so on. Suppose that the objective criterion is a quadratic function, with equality constraints representing the system dynamics, hyperboxes and binary constraints on the inputs. Branch and bound method can be used to solve the centralized hybrid MPC problem, which can be cast in MIQP formulation. A large number of subsystems and the corresponding binary inputs correspond to a huge decision space, which will lead to an excessive number of branches. To overcome this difficulty, the idea is to make use of the tighter bounds provided by the SDP relaxation compared to the QP relaxation so as to reduce the size of the tree that has to be explored. However, as shown in [4], the price for tighter bounds comes along with a higher computational burden. The main contribution in this paper is to define a mechanism that compensates for this drawback, by distributing the computation load of calculating bounds over the subsystems. The distributed structure can be obtained using dual decomposition to relax the complicating constraints. Since the SDP relaxation problem is already convexified, strong duality conditions are ensured. So, the duality gap is absent and convergence is guaranteed.

The paper is organized as follows. Section II, introduces the system under study and formulates the centralized hybrid MPC

[1]Amir Firooznia and Bart De Schutter are with the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands {a.firooznia,b.deschutter}@tudelft.nl

[2]Romain Bourdais is with the CentraleSupélec/IETR, Cesson-Sévigné, France romain.bourdais@centralesupelec.fr

problem. Moreover, the centralized QP and SDP relaxations of the original problem are obtained for later comparison. In Section III, the SDP relaxation is distributed over the subsystems via a dual decomposition method and distributed algorithms are presented to find lower and upper bounds. In Section IV, the effectiveness of the proposed algorithm is illustrated with a case study.

II. Problem Statement

A. Interconnected Dynamical Hybrid Model

We consider the following interconnected system:

$$x_1(k) = A_{11}x_1(k-1) + \sum_{i\in\mathcal{I}} B_{1i}u_i(k) \tag{1}$$

$$x_j(k) = A_{jj}x_j(k-1) + B_{jj}u_j(k) \quad \text{for} \quad j \in \mathcal{I}\setminus\{1\} \tag{2}$$

$$x_i(0) = x_{i,0} \quad \text{for} \quad i \in \mathcal{I} \tag{3}$$

$$u_1(k) \in \bigtimes_{1,\dots,m_1} [u_{1,\min}, u_{1,\max}] \tag{4}$$

$$u_j \in \{0,1\}^{m_j} \quad \text{for} \quad j \in \mathcal{I}\setminus\{1\} \tag{5}$$

where $\mathcal{I} := \{1,\dots,N_{\text{sys}}\}$, N_{sys} is the number of subsystems, $x_i(k) \in \mathbb{R}^{n_i}$ for $i \in \mathcal{I}$ and $u_{1,\min}, u_{1,\max} \in \mathbb{R}$. Here, $H = \bigtimes_{1,\dots,\ell}[a,b]$ defines a hyperbox in $\mathbb{R}^\ell$ over the interval $[a,b]$, i.e., $H = [a,b] \times \dots \times [a,b]$. In the given formulation, the first subsystem with mixed-integer inputs can be considered as a producer and the rest of subsystems can be treated as consumers. The interconnection among the subsystems is due to the nature of producer, which interacts with consumers (see [20]). The corresponding optimization problem can be formulated within the framework of mixed-integer problems.

B. Centralized MIQP Problem

Let us define:

$$\tilde{X}_i := \begin{bmatrix} x_i(1) \\ \vdots \\ x_i(N_\text{p}) \end{bmatrix}, \; \tilde{U}_i := \begin{bmatrix} u_i(1) \\ \vdots \\ u_i(N_\text{p}) \end{bmatrix} \text{ for } i \in \mathcal{I}, \tag{6}$$

$$\tilde{X} := \begin{bmatrix} \tilde{X}_1 \\ \vdots \\ \tilde{X}_{N_\text{sys}} \end{bmatrix}, \; \tilde{U} := \begin{bmatrix} \tilde{U}_1 \\ \vdots \\ \tilde{U}_{N_\text{sys}} \end{bmatrix}, \tag{7}$$

with N_p the prediction horizon. We consider the following quadratic cost function designed for reference tracking while penalizing the energy consumption of the system:

$$\begin{aligned} J(\tilde{X},\tilde{U}) := \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} &\frac{1}{2}(x_i(k) - x_{i,\text{ref}}(k))^T Q_{x,i}(x_i(k) - x_{i,\text{ref}}(k)) \\ &+ \frac{1}{2}u_i(k)^T Q_{u,i} u_i(k), \end{aligned} \tag{8}$$

where $\mathcal{K} := \{1,\dots,N_\text{p}\}$, $Q_{x,i} \in \mathbb{S}_+^{n_i}$ and $Q_{u,i} \in \mathbb{S}_{++}^{m_i}$ for $i \in \mathcal{I}$ with $\mathbb{S}_+^n$ and $\mathbb{S}_{++}^n$ the sets of positive semi-definite and positive definite symmetric matrices of size $n \times n$, respectively. Then, we formulate the centralized MIQP problem as follows:

$$\begin{aligned} \min_{\tilde{X}_\text{MIQP},\tilde{U}_\text{MIQP}} \quad & J(\tilde{X}_\text{MIQP},\tilde{U}_\text{MIQP}) \\ \text{s.t.} \quad & (1)-(5) \end{aligned} \tag{9}$$

with the optimal value $J(\tilde{X}^\star_\text{MIQP},\tilde{U}^\star_\text{MIQP})$. Suppose $N := \sum_{i\in\mathcal{I}} n_i$ and $M := \sum_{i\in\mathcal{I}} m_i$. Here, the total number of decision variables is $(N+M)N_\text{p}$, of which $(M-m_1)N_\text{p}$ are binary.

C. QP Relaxation of the Centralized Problem

The QP relaxation of constraint (5) is equivalent to:

$$u_{j,\text{QP}} \in \bigtimes_{1,\dots,m_j} [0,1] \quad \text{for} \quad j \in \mathcal{I}\setminus\{1\} \tag{10}$$

Define $\tilde{X}_{i,\text{QP}}$, $\tilde{X}_\text{QP}$, $\tilde{U}_{i,\text{QP}}$, and $\tilde{U}_\text{QP}$ similar to (6)-(7). So, problem (9) can be relaxed for the following QP problem:

$$\begin{aligned} \min_{\tilde{X}_\text{QP},\tilde{U}_\text{QP}} \quad & J(\tilde{X}_\text{QP},\tilde{U}_\text{QP}) \\ \text{s.t.} \quad & (1)-(4),\ (10) \end{aligned} \tag{11}$$

with the optimal value $J(\tilde{X}^\star_\text{QP},\tilde{U}^\star_\text{QP})$. Here, the total number of decision variables is $(N+M)N_\text{p}$, of which none is binary. Obviously, the optimal value of the relaxed problem gives a lower bound for the original MIQP problem, i.e., we have

$$J(\tilde{X}^\star_\text{QP},\tilde{U}^\star_\text{QP}) \le J(\tilde{X}^\star_\text{MIQP},\tilde{U}^\star_\text{MIQP}) \tag{12}$$

D. SDP Relaxation of the Centralized Problem

Now we define the following structure for the inputs:

$$\tilde{U}_\text{c} := \tilde{U}_1 \;,\; \tilde{U}_\text{b} := \begin{bmatrix} \tilde{U}_2 \\ \vdots \\ \tilde{U}_{N_\text{sys}} \end{bmatrix} \Rightarrow \tilde{U} := \begin{bmatrix} \tilde{U}_\text{c} \\ \tilde{U}_\text{b} \end{bmatrix} \tag{13}$$

where $\tilde{U}_\text{c}$ is the vector of real inputs (corresponding to the producer) while $\tilde{U}_\text{b}$ is the vector of binary inputs (corresponding to the consumers). Now, consider $\tilde{V} = \begin{bmatrix}\tilde{V}_\text{c}^T & \tilde{V}_\text{b}^T\end{bmatrix}^T$ in which $\tilde{V}_\text{c}$ has the same nature as in $\tilde{U}_\text{c}$ while the elements in the vector $\tilde{V}_\text{b}$ can take any real value between 0 and 1. Moreover, consider the following new cost function:

$$\begin{aligned} \bar{J}(\tilde{X},\tilde{V},\mathcal{W}) :=& \frac{1}{2}(\tilde{X}-\tilde{X}_\text{ref})^T\tilde{Q}_x(\tilde{X}-\tilde{X}_\text{ref}) \\ &+ \frac{1}{2}\tilde{V}_\text{c}^T\tilde{Q}_{u,1}\tilde{V}_\text{c} + \frac{1}{2}\text{trace}(\tilde{Q}_{u,\text{b}}\mathcal{W}) \end{aligned} \tag{14}$$

where we have:

$$\tilde{Q}_{x,i} := \text{diag}(\underbrace{Q_{x,i},\dots,Q_{x,i}}_{N_\text{p}\ \text{times}}) \text{ for } i \in \mathcal{I} \tag{15}$$

$$\tilde{Q}_{u,i} := \text{diag}(\underbrace{Q_{u,i},\dots,Q_{u,i}}_{N_\text{p}\ \text{times}}) \text{ for } i \in \mathcal{I} \tag{16}$$

$$\tilde{Q}_x := \text{diag}(\tilde{Q}_{x,1},\dots,\tilde{Q}_{x,N_\text{sys}}) \tag{17}$$

$$\tilde{Q}_{u,\text{b}} := \text{diag}(\tilde{Q}_{u,2},\dots,\tilde{Q}_{u,N_\text{sys}}) \tag{18}$$

where the matrix variable $\mathcal{W} \in \mathbb{S}_+^{n_w}$ with $n_w = n_\text{b}$ satisfies the following conditions:

$$\text{diag}(\mathcal{W}) = \tilde{V}_\text{b} \tag{19}$$

$$\begin{bmatrix} \mathcal{W} & \tilde{V}_\text{b} \\ \tilde{V}_\text{b}^T & 1 \end{bmatrix} \succeq 0 \tag{20}$$

where the off-diagonal elements of $\mathcal{W}$ are extra added variables. Assume $\mathcal{W}$ has the following structure:

$$\begin{bmatrix} W_2(1) & \star & \star & \star & \star & \star & \star \\ \star & \ddots & \star & \star & \star & \star & \star \\ \star & \star & W_2(N_\mathrm{p}) & \star & \star & \star & \star \\ \star & \star & \star & \ddots & \star & \star & \star \\ \star & \star & \star & \star & W_{N_\mathrm{sys}}(1) & \star & \star \\ \star & \star & \star & \star & \star & \ddots & \star \\ \star & \star & \star & \star & \star & \star & W_{N_\mathrm{sys}}(N_\mathrm{p}) \end{bmatrix} \tag{21}$$

where $W_j(k) \in \mathbb{S}_+^{m_j}$ for $j \in \mathcal{I} \setminus \{1\}$ and $k \in \mathcal{K}$ and $\star$ is used to denote the remaining off-diagonal (and symmetric) variables within $\mathcal{W}$. Since $\tilde{Q}_{u,\mathrm{b}}$ in (14) is block-diagonal, $\mathrm{trace}(\tilde{Q}_{u,\mathrm{b}}\mathcal{W})$ will only include the corresponding block-diagonal elements of $\mathcal{W}$ and the off-diagonal blocks of $\mathcal{W}$ will not be present in the objective function. Therefore, we can rewrite the last element in the objective function (14) in terms of only block-diagonal elements of $\mathcal{W}$ as follows:

$$\mathrm{trace}(\tilde{Q}_{u,\mathrm{b}}\mathcal{W}) = \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{I}\setminus\{1\}} \mathrm{trace}(\tilde{Q}_{u,j} W_j(k)) \tag{22}$$

Thus, the off-diagonal elements (denoted by $\star$) are not present in the cost (14) and the cost becomes decomposable. Moreover, based on Lemma 1 in appendix such off-diagonal elements resulting in a positive semi-definite $\mathcal{W}$ exist if and only if (ii) in the lemma is satisfied and therefore (20) can be decomposed into smaller LMIs only containing the diagonal elements.

Hence, we formulate the relaxed SDP problem as follows:

$$\min_{\tilde{X},\tilde{V},\mathcal{W}} \quad \bar{J}(\tilde{X},\tilde{V},\mathcal{W}) \tag{23}$$

$$\text{s.t. } x_1(k) = A_{11}x_1(k-1) + \sum_{i\in\mathcal{I}} B_{1i}v_i(k) \tag{24}$$

$$x_j(k) = A_{jj}x_j(k-1) + B_{jj}v_j(k) \text{ for } j \in \mathcal{I}\setminus\{1\} \tag{25}$$

$$x_i(0) = x_{i,0} \quad \text{for} \quad i \in \mathcal{I} \tag{26}$$

$$v_1(k) \in \bigtimes_{1,\dots,m_1} [u_{1,\min}, u_{1,\max}] \tag{27}$$

$$\mathrm{diag}(W_j(k)) = v_j(k) \tag{28}$$

$$\begin{bmatrix} W_j(k) & v_j(k) \\ (v_j(k))^T & 1 \end{bmatrix} \succeq 0 \quad \text{for} \quad j \in \mathcal{I}\setminus\{1\} \tag{29}$$

with the optimal value $\bar{J}(\tilde{X}^\star_{\mathrm{SDP}}, \tilde{V}^\star_{\mathrm{SDP}}, \mathcal{W}^\star_{\mathrm{SDP}})$. Here, the total number of decision variables is $(N + M + \sum_{j\in\mathcal{I}\setminus\{1\}} (m_j^2 - m_j)/2)N_\mathrm{p}$, of which none is binary and which grows linearly with N_p.

Exploiting the binary nature of the decision variables stacked in $\tilde{U}_\mathrm{b}$, it can be shown that $J(\tilde{X}^\star_{\mathrm{SDP}}, \tilde{V}^\star_{\mathrm{SDP}}) \le \bar{J}(\tilde{X}^\star_{\mathrm{SDP}}, \tilde{V}^\star_{\mathrm{SDP}}, \mathcal{W}^\star_{\mathrm{SDP}})$ (for a complete treatment see Section 3.3 in [7]). Moreover, since the pair $(\tilde{X}^\star_{\mathrm{QP}}, \tilde{U}^\star_{\mathrm{QP}})$ is the solution for (11), $J(\tilde{X}^\star_{\mathrm{QP}}, \tilde{U}^\star_{\mathrm{QP}}) \le J(\tilde{X}^\star_{\mathrm{SDP}}, \tilde{V}^\star_{\mathrm{SDP}})$ necessarily holds. Note that the number of decision variables of the SDP problem is larger compared to the QP relaxation technique (with the difference $(\sum_{j\in\mathcal{I}\setminus\{1\}} (m_j^2 - m_j)/2)N_\mathrm{p}$). Moreover, we have:

$$J(\tilde{X}^\star_{\mathrm{QP}}, \tilde{U}^\star_{\mathrm{QP}}) \le \bar{J}(\tilde{X}^\star_{\mathrm{SDP}}, \tilde{V}^\star_{\mathrm{SDP}}, \mathcal{W}^\star_{\mathrm{SDP}}) \le J(\tilde{X}^\star_{\mathrm{MIQP}}, \tilde{U}^\star_{\mathrm{MIQP}}) \tag{30}$$

and due to the latter inequality SDP relaxation yields a tighter lower bound for the MIQP problem. Moreover, as we will explain in the next section, the complicating constraint (24), can be relaxed via Lagrangian relaxation to decompose the computation load into each subsystem.

III. Distributed Algorithm to Find Bounds using Dual Decomposition

A. Lagrangian Dual Problem and Distributed Structure

Our goal is to distribute the primal SDP relaxed optimization problem (23)-(29) into local subproblems. Note that the objective function can be decomposed into a sum of local functions, but there exists global complicating constraint (24). We will apply the Lagrangian relaxation technique to separate the complicating constraint. Let us define the dual variable $\Lambda = [\lambda^T(1), \dots, \lambda^T(N_\mathrm{p})]^T$, which is also called the Lagrangian multiplier vector associated with (24) and consider the Lagrangian function associated to problem (23)-(29), which can be partitioned into multiple sub-functions as follows:

$$\begin{aligned}
&\mathcal{L}(\tilde{X},\tilde{V},\mathcal{W};\Lambda) := \bar{J}(\tilde{X},\tilde{V},\mathcal{W}) \\
&\quad + \sum_{k\in\mathcal{K}} \lambda^T(k)\Big(x_1(k) - A_{11}x_1(k-1) - B_{11}v_1(k) \\
&\qquad\qquad - \sum_{j\in\mathcal{I}\setminus\{1\}} B_{1j}v_j(k)\Big) \\
&= \underbrace{\frac{1}{2}(\tilde{X}_1 - \tilde{X}_{1,\mathrm{ref}})^T \tilde{Q}_{x,1}(\tilde{X}_1 - \tilde{X}_{1,\mathrm{ref}}) + \frac{1}{2}\tilde{V}_1^T\tilde{Q}_{u,1}\tilde{V}_1 + \sum_{k\in\mathcal{K}} \lambda^T(k)\left(x_1(k) - A_{11}x_1(k-1) - B_{11}v_1(k)\right)}_{=:\mathcal{L}_{\mathrm{SDP},1}} \\
&\quad + \sum_{j\in\mathcal{I}\setminus\{1\}} \underbrace{\Bigg(\frac{1}{2}(\tilde{X}_j - \tilde{X}_{j,\mathrm{ref}})^T\tilde{Q}_{x,j}(\tilde{X}_j - \tilde{X}_{j,\mathrm{ref}}) + \sum_{k\in\mathcal{K}} \frac{1}{2}\mathrm{trace}\left(\tilde{Q}_{u,j}W_j(k)\right) - \lambda^T(k)B_{1j}v_j(k)\Bigg)}_{=:\mathcal{L}_{\mathrm{SDP},j}} \\
&= \mathcal{L}_{\mathrm{SDP},1}(\tilde{X}_1,\tilde{V}_1;\Lambda) + \sum_{j\in\mathcal{I}\setminus\{1\}} \mathcal{L}_{\mathrm{SDP},j}(\tilde{X}_j,\tilde{V}_j,W_j;\Lambda)
\end{aligned} \tag{31}$$

in which the distributed local j-th Lagrangian function is denoted by $\mathcal{L}_{\mathrm{SDP},j}(\tilde{X}_j,\tilde{V}_j,W_j;\Lambda)$. The structure exploited is a classic one for distributed optimization [21]; it is composed of N_sys parallel MPC optimization corresponding to each subsystem and a coordinator for ensuring the consensus. The structure is illustrated in Figure 1. In this way, we obtain a decomposed problem where each subsystem (producer or consumer) has to solve its own optimization for some Λ and then negotiate the value of Lagrangian multiplier with other agents through the coordinator to update Λ. In other words, for a given Lagrangian multiplier Λ, subsystems have to solve their own optimization problem (with respect to the corresponding local Lagrangian functions) simultaneously.

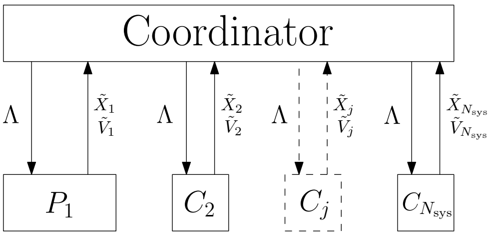

Fig. 1. Distributed structure for N_{sys} subsystems where P_1 corresponds to the producer as the first subsystem and C_j denotes the consumers.

For the producer the following minimization subproblem has to be solved with respect to the primal variables $\tilde{X}_1, \tilde{V}_1$:

$$\begin{aligned} \Phi_1(\Lambda) := \min_{\tilde{X}_1, \tilde{V}_1} \quad & \mathcal{L}_{\text{SDP},1}(\tilde{X}_1, \tilde{V}_1; \Lambda) \\ \text{s.t.} \quad & x_1(0) = x_{1,0} \\ & v_1(k) \in \bigtimes_{1,\dots,m_1} [u_{1,\min}, u_{1,\max}] \end{aligned} \tag{32}$$

and each consumer $j \in \mathcal{I} \setminus \{1\}$ needs to solve its own minimization problem with respect to the primal variables $\tilde{X}_j, \tilde{V}_j, W_j$:

$$\begin{aligned} \Phi_j(\Lambda) := \min_{\tilde{X}_j, \tilde{V}_j, W_j} \quad & \mathcal{L}_{\text{SDP},j}(\tilde{X}_j, \tilde{V}_j, W_j; \Lambda) \\ \text{s.t.} \quad & x_j(k) = A_{jj} x_j(k-1) + B_{jj} v_j(k), \\ & x_j(0) = x_{j,0}, \\ & \operatorname{diag}(W_j(k)) = v_j(k) \\ & \begin{bmatrix} W_j(k) & v_j(k) \\ (v_j(k))^T & 1 \end{bmatrix} \succeq 0 \end{aligned} \tag{33}$$

The following Lagrangian dual problem has to be solved to obtain the optimal dual variable $\Lambda^\star$:

$$\max_{\Lambda} \quad \mathbf{\Phi}(\Lambda) \tag{34}$$

where the overall Lagrangian dual function is denoted by

$$\mathbf{\Phi}(\Lambda) = \sum_{i \in \mathcal{I}} \Phi_i(\Lambda) \tag{35}$$

From the standard duality theory, and due to the fact that strong duality holds for the SDP convex problem (23)-(29), solving the corresponding optimization is equivalent to solving the unconstrained maximization in (34). One common approach to solve (34) is to use a subgradient method [21]. The resulting distributed algorithm is described in Algorithm 1. In this case (with relaxed variables), the iteration algorithm will converge to the maximum of the dual function. Therefore, the convergence to the global solution of the SDP problem (23)-(29) is ensured due to the convexity of the problem. Moreover, in Algorithm 1, the distributed part is realized in the sense that subsystems will solve their minimization problems in parallel at each iteration. The rest of the procedure is ensured by the coordinator that has a central role but that does not perform any optimization.

Initialization: $p \leftarrow 1$, $\Lambda_p \leftarrow 0$;
while *Convergence criterion is NOT satisfied* **do**
  $(\tilde{X}^\star_{1,p}, \tilde{V}^\star_{1,p}) \leftarrow$ Solve $\Phi_1(\Lambda_p)$;
  for $j \in \mathcal{I} \setminus \{1\}$ **do**
    $(\tilde{X}^\star_{j,p}, \tilde{V}^\star_{j,p}, W^\star_{j,p}) \leftarrow$ Solve $\Phi_j(\Lambda_p)$;
  end
  Send $\tilde{X}^\star_{1,p}, \tilde{V}^\star_{1,p}, \dots, \tilde{V}^\star_{N_{\text{sys}},p}$ to the coordinator agent;
  Evaluate convergence criterion;
  if *Convergence criteria is satisfied* **then**
    $\Lambda^\star \leftarrow \Lambda_p$;
    $(\tilde{X}^\star_1, \tilde{V}^\star_1) \leftarrow (\tilde{X}^\star_{1,p}, \tilde{V}^\star_{1,p})$;
    $(\tilde{X}^\star_j, \tilde{V}^\star_j, W^\star_j) \leftarrow (\tilde{X}^\star_{j,p}, \tilde{V}^\star_{j,p}, W^\star_{j,p})$ for $j \in \mathcal{I} \setminus \{1\}$;
    $\mathbf{\Phi}(\Lambda^\star) \leftarrow \sum_{i \in \mathcal{I}} \Phi_i(\Lambda^\star)$;
    exit while loop;
  end
  $\Lambda_{p+1} \leftarrow$ Update Λ_p using subgradient method;
  $p \leftarrow p + 1$;
end
return $\Lambda^\star$, $\mathbf{\Phi}(\Lambda^\star)$, $\tilde{X}^\star$, $\tilde{V}^\star$, $\mathcal{W}^\star$;

Algorithm 1: Distributed solution for the Lagrangian dual problem.

B. Assigning the Lower Bound

Let us denote the primal and dual variable tuple obtained by Algorithm 1 after iteration p with $(\tilde{X}_{\text{DSDP},p}, \tilde{V}_{\text{DSDP},p}, \mathcal{W}_{\text{DSDP},p}; \Lambda_{\text{DSDP},p})$ and similarly denote the optimal primal and dual variable tuple by $(\tilde{X}^\star_{\text{DSDP},p}, \tilde{V}^\star_{\text{DSDP},p}, \mathcal{W}^\star_{\text{DSDP},p}; \Lambda^\star_{\text{DSDP},p})$. Due to the convergence property discussed above, can write:

$$\begin{aligned} \bar{J}(\tilde{X}_{\text{DSDP},p}, \tilde{V}_{\text{DSDP},p}, \mathcal{W}_{\text{DSDP},p}) &\geq \bar{J}(\tilde{X}^\star_{\text{SDP}}, \tilde{V}^\star_{\text{SDP}}, \mathcal{W}^\star_{\text{SDP}}) \\ \bar{J}(\tilde{X}^\star_{\text{DSDP}}, \tilde{V}^\star_{\text{DSDP}}, \mathcal{W}^\star_{\text{DSDP}}) &= \bar{J}(\tilde{X}^\star_{\text{SDP}}, \tilde{V}^\star_{\text{SDP}}, \mathcal{W}^\star_{\text{SDP}}) \end{aligned} \tag{36}$$

Moreover, we have:

$$\begin{aligned} \mathbf{\Phi}(\Lambda_{\text{DSDP},p}) &\leq \bar{J}(\tilde{X}^\star_{\text{SDP}}, \tilde{V}^\star_{\text{SDP}}, \mathcal{W}^\star_{\text{SDP}}) \\ \mathbf{\Phi}(\Lambda^\star_{\text{DSDP}}) &= \bar{J}(\tilde{X}^\star_{\text{SDP}}, \tilde{V}^\star_{\text{SDP}}, \mathcal{W}^\star_{\text{SDP}}) \end{aligned} \tag{37}$$

Thus, using (30) and (37) we obtain

$$J(\tilde{X}^\star_{\text{QP}}, \tilde{U}^\star_{\text{QP}}) \leq \mathbf{\Phi}(\Lambda^\star_{\text{DSDP}}) \leq J(\tilde{X}^\star_{\text{MIQP}}, \tilde{U}^\star_{\text{MIQP}}) \tag{38}$$

In other words, $\mathbf{\Phi}(\Lambda^\star_{\text{DSDP}})$ yields a tighter lower bound - compared to the one obtained using QP relaxation - for the centralized problem, and it can be computed using a distributed structure.

C. Assigning an Upper Bound

To assign an upper bound to the centralized MIQP problem (9) at each node within a branch and bound method, one simple method is to assign random binary inputs for consumers and to solve the resulting QP problem for the producer (see (42) in appendix). Any answer found this way will provide an upper bound. This procedure can be repeated several times and we can use the lowest upper bound found. However, when the number of subsystems (and as a result the number of binary decision variables) increases, random selection of variables within the search space is less likely to provide a good upper bound, especially during the first branching stage where no preliminary warm start exists.

However, a potentially more efficient approach is to use the intermediate obtained results in Algorithm 1. To this aim, at each iteration p, while the convergence criterion is not yet satisfied, the SDP relaxed variables $V_{j,p}$ for $j \in \mathcal{I} \setminus \{1\}$ can be rounded to the corresponding binary variable 0 or 1, to form a vector denoted by $\boldsymbol{V}_{j,p}$, which can be used in the producer problem as the input from the consumers side. Then, the resulting QP problem can be solved to give an upper bound. We compare the obtained bound with the one obtained in the previous iteration $p-1$ and keep the minimum as the upper bound, which makes sure the upper bound converge to some steady state value. This algorithm can be implemented within Algorithm 1 with some modifications to form a more complete Algorithm 2. The output of Algorithm 2 is the lower bound $\boldsymbol{\Phi}(\Lambda^\star_{\text{DSDP}})$ and upper bound $\Psi^\star_{\text{DSDP}}$ found via distributed structure.

Initialization: $p \leftarrow 1$, $\Lambda_p \leftarrow 0$, $\Psi^\star \leftarrow \infty$;
while *Convergence criterion is NOT satisfied* **do**
  $(\tilde{X}^\star_{1,p}, \tilde{V}^\star_{1,p}) \leftarrow$ Solve $\Phi_1(\Lambda_p)$;
  for $j \in \mathcal{I} \setminus \{1\}$ **do**
    $(\tilde{X}^\star_{j,p}, \tilde{V}^\star_{j,p}, W^\star_{j,p}) \leftarrow$ Solve $\Phi_j(\Lambda_p)$;
    $\boldsymbol{V}_{j,p} \leftarrow$ Round$(\tilde{V}^\star_{j,p})$
  end
  $(\tilde{X}^\star, \tilde{U}^\star_{\text{c}}) \leftarrow$ Solve (42) with $\hat{J}_{\text{P}}(\tilde{X}, \tilde{U}_{\text{c}}, \boldsymbol{V}_{j,p})$;
  $\psi_p \leftarrow \hat{J}_{\text{P}}(\tilde{X}^\star, \tilde{U}^\star_{\text{c}}, \boldsymbol{V}_{j,p})$;
  $\Psi^\star \leftarrow \min(\psi_p, \Psi^\star)$;
  Send $\tilde{X}^\star_{1,p}, \tilde{V}^\star_{1,p}, \ldots, \tilde{V}^\star_{N_{\text{sys}},p}$ to the coordinator agent;
  Evaluate convergence criterion;
  if *Convergence criterion is satisfied* **then**
    $\Lambda^\star \leftarrow \Lambda_p$;
    $(\tilde{X}^\star_1, \tilde{V}^\star_1) \leftarrow (\tilde{X}^\star_{1,p}, \tilde{V}^\star_{1,p})$;
    $(\tilde{X}^\star_j, \tilde{V}^\star_j, W^\star_j) \leftarrow (\tilde{X}^\star_{j,p}, \tilde{V}^\star_{j,p}, W^\star_{j,p})$ for $j \in \mathcal{I} \setminus \{1\}$;
    $\boldsymbol{\Phi}(\Lambda^\star) \leftarrow \sum_{i \in \mathcal{I}} \Phi_i(\Lambda^\star)$;
    exit while loop;
  end
  $\Lambda_{p+1} \leftarrow$ Update Λ_p using subgradient method;
  $p \leftarrow p + 1$;
end
return $\Lambda^\star$, $\boldsymbol{\Phi}(\Lambda^\star)$, $\Psi^\star$, $\tilde{X}^\star$, $\tilde{V}^\star$, $\mathcal{W}^\star$;

Algorithm 2: Modified distributed solution for the Lagrangian dual problem.

IV. Simulation Results

To illustrate our approach, let us consider a case study related to the producer and consumers case. Table I lists the parameters of this case study. All system matrices are built by the MATLAB random system generator `drss` and the weighing matrices are considered as full positive (semi)-definite.

In the first simulation, the optimal cost functions for the centralized MIQP problem, the QP relaxation, and the SDP relaxation problems are compared as tabulated in Table II. As can be seen the results conform to inequality (30).

N_{sys}	3
$n_1 = n_2 = n_3$	2
m_1	1
$m_2 = m_3$	3
$[u_{1,\min}, u_{1,\max}]$	$[-2, 2]$
N_{p}	3

TABLE I
Parameters for simulation.

Method	Optimal cost
original MIQP	9.0788
QP relaxation	7.1960
SDP relaxation	7.9701

TABLE II
Simulation results for Section II.

Next, Algorithm 1 is used to find a solution for the Lagrangian dual problem of Section III and the results are depicted in Figure 2. As can be seen, relations (36) and (37) both hold and the dual Lagrangian function converges to the lower bound obtained by centralized SDP.

In the last simulation, Algorithm 2 is used to produce the lower and upper bounds for the original centralized MIQP problem and the results are illustrated in Figure 3. Moreover, in this special case, the upper bound has converged to the centralized MIQP solution, which does not necessarily hold in general.

V. Conclusions

A distributed procedure has been proposed to determine a lower bound and an upper bound of the MIQP optimization problem corresponding to a class of interconnected hybrid systems in order to limit the search space in branch and bound method. The lower bound is obtained using a SDP relaxation of the original problem, which is solved using a dual decomposition approach. As the SDP relaxation defines a strong convex problem, the convergence of the gradient method is ensured. The upper bound is obtained by using the rounded values of the relaxed problem. Simulation results for a case study show how the algorithms work.

In this work, consumers can only interact with the producer. A more general setting involving consumers inter-

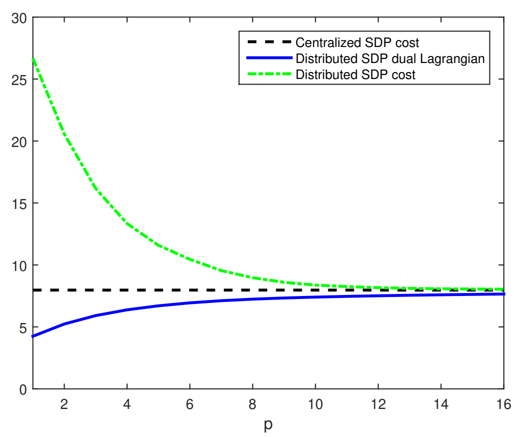

Fig. 2. Evolution of the distributed SDP optimization problem cost and the corresponding dual Lagrangian value. Note that convergence to the centralized solution occurs.

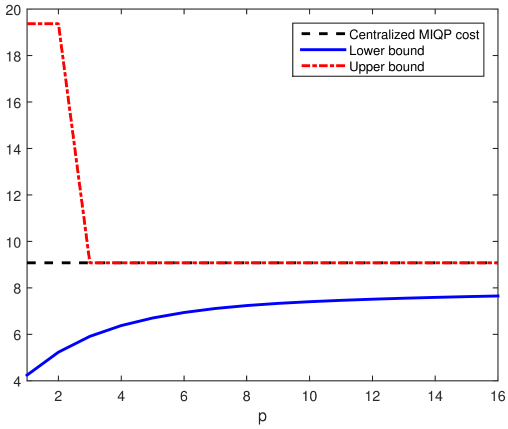

Fig. 3. Evolution of the lower and upper bounds found by Algorithm 2 and comparison with the centralized solution.

connected to each other in a general network topology will be considered in future work. Another direction will be to investigate the upper bound evolution more closely to improve the upper bound in the provided algorithm and to determine the tightness of the provided bounds more precisely. We will also assess the efficiency of the approach. In addition, we will develop a branch and bound method based on the bounds obtained in this work.

APPENDIX

Lemma 1 ([7]): For given vectors $p_i \in \mathbb{R}^{\dim(p_i)}$ and matrices $P_{ii} \in \mathbb{S}_+^{\dim(p_i)}$ for $i,j = 1,\ldots,q$ and $i \neq j$, the following statements are equivalent:

$$(i) \exists P_{ij} \in \mathbb{R}^{\dim(p_i)\times\dim(p_j)} \text{ for } i,j=1,\ldots,q,\ i\neq j: \begin{bmatrix} P_{11} & \ldots & P_{1q} & p_1 \\ \vdots & \ddots & \vdots & \vdots \\ P_{1q}^T & \ldots & P_{qq} & p_q \\ p_1^T & \ldots & p_q^T & 1 \end{bmatrix} \succeq 0 \quad (39)$$

$$(ii) \begin{bmatrix} P_{11} & p_1 \\ p_1^T & 1 \end{bmatrix} \succeq 0, \ldots, \begin{bmatrix} P_{qq} & p_q \\ p_q^T & 1 \end{bmatrix} \succeq 0 \quad (40)$$

Definition 1: Using (13), we can rewrite the original cost function in (8) as $J(\tilde{X},\tilde{U}) = \hat{J}(\tilde{X},\tilde{U}_{\mathrm{c}},\tilde{U}_{\mathrm{b}})$. Then, for a given consumer input $\tilde{U}_{\mathrm{b}} = [\tilde{U}_{\mathrm{b},2}^T,\ldots,\tilde{U}_{\mathrm{b},N_{\mathrm{sys}}}^T]^T$, define the producer cost function as follows:

$$\begin{aligned}\hat{J}(\tilde{X},\tilde{U}_{\mathrm{c}},\tilde{U}_{\mathrm{b}}) = &\frac{1}{2}(\tilde{X}_1-\tilde{X}_{1,\mathrm{ref}})^T\tilde{Q}_{x,1}(\tilde{X}_1-\tilde{X}_{1,\mathrm{ref}})+ \\ &+\frac{1}{2}\tilde{U}_{\mathrm{c}}^T\tilde{Q}_{u,1}\tilde{U}_{\mathrm{c}} \\ &+\sum_{j\in\mathcal{I}\setminus\{1\}}\frac{1}{2}(\tilde{X}_j-\tilde{X}_{j,\mathrm{ref}})^T\tilde{Q}_{x,j}(\tilde{X}_j-\tilde{X}_{j,\mathrm{ref}}) \\ &+\frac{1}{2}\tilde{U}_{\mathrm{b},j}^T\tilde{Q}_{u,j}\tilde{U}_{\mathrm{b},j}\end{aligned} \quad (41)$$

Then for a given $\tilde{U}_{\mathrm{b}}$ the producer QP problem can be formulated as follows:

$$\begin{aligned}\min_{\tilde{X},\tilde{U}_{\mathrm{c}}} \quad & \hat{J}_{\mathrm{P}}(\tilde{X},\tilde{U}_{\mathrm{c}},\tilde{U}_{\mathrm{b}}) \\ \text{s.t.} \quad & (1)-(4)\end{aligned} \quad (42)$$

with $\hat{J}_{\mathrm{P}}$ used to denote the cost of the producer problem.

ACKNOWLEDGMENT

Work supported by the Dutch Technology Foundation STW through the Robust CPS program (project 12693), by Ammodo via the Van Gogh grant VGP.14/47, and by Rennes Métropole grant.

REFERENCES

[1] E. H. Camacho, "Model predictive control techniques for hybrid systems," *Annual Reviews in Control*, vol. 34, no. 1, pp. 21–31, Apr. 2010.

[2] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer Programming*. Springer International Publishing, 2014.

[3] V. T. Paschos, *Concepts of Combinatorial Optimization*, 2nd ed. Wiley, 2014.

[4] D. Axehill, L. Vandenberghe, and A. Hansson, "Convex relaxations for mixed integer predictive control," *Automatica*, vol. 46, no. 9, pp. 1540–1545, Jan. 2010.

[5] P. A. Parrilo and S. Lall, "Semidefinite programming relaxations and algebraic optimization in control," *European Journal of Control*, vol. 9, no. 2-3, pp. 307–321, 2003.

[6] A. Billionnet and S. Elloumi, "Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem," *Mathematical Programming*, vol. 109, no. 1, pp. 55–68, Jan. 2007.

[7] D. Axehill, L. Vandenberghe, and A. Hansson, "On relaxations applicable to model predictive control for systems with binary control signals," in *Preprints of the 7th IFAC Symposium on Nonlinear Control Systems*, Pretoria, South Africa, 2007, pp. 200–205.

[8] A. J. Conejo, E. Castillo, R. Mínguez, and R. García-Bertrand, *Decomposition Techniques in Mathematical Programming*, 1st ed. Springer, 2006.

[9] J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho, "Distributed model predictive control based on a cooperative game," *Optimal Control Applications and Methods*, vol. 32, no. 2, pp. 153–176, 2011.

[10] Y. Zhang and S. Li, "Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes," *Journal of Process Control*, vol. 17, no. 1, pp. 37–50, Jan. 2007.

[11] 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, Feb. 2002.

[12] 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, no. 3, pp. 6–15, 2009.

[13] D. P. Bertsekas, "Separable dynamic programming and approximate decomposition methods," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 911–916, May 2007.

[14] J. B. Rawlings and B. T. Stewart, "Coordinating multiple optimization-based controllers: New opportunities and challenges," *Journal of Process Control*, vol. 18, no. 9, pp. 839–845, 2008.

[15] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1192–1206, Nov. 2008.

[16] J. M. Maestre and R. R. Negenborn, *Distributed Model Predictive Control Made Easy*, 1st ed. Springer, 2014.

[17] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, Dec. 2014.

[18] R. Bourdais, H. Guéguen, and A. Belmiloudi, "Distributed model predictive control for a class of hybrid system based on lagrangian relaxation," in *Proceedings of the 4th IFAC Conference on Analysis and Design of Hybrid Systems*, Eindhoven, The Netherlands, 2012, pp. 46–51.

[19] A. Rantzer, "Dynamic dual decomposition for distributed control," in *the 2009 Proceedings of American Control Conference*, St. Louis, USA, 2009, pp. 884–888.

[20] E. Herrera, R. Bourdais, and H. Guéguen, "Predictive and interactive controllers for solar absorption cooling systems in buildings," *Journal of Process Control*, vol. 24, no. 6, pp. 836–845, Jun. 2014.

[21] D. P. Bertsekas, *Convex Optimization Algorithms*, 1st ed. Athena Scientific, 2015.