2022 30th Mediterranean Conference on Control and Automation (MED)
June 28 - July 1, 2022. Athens, Greece

2022 30th Mediterranean Conference on Control and Automation (MED) | 978-1-6654-0673-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/MED54222.2022.9837180

Hierarchical Distributed Model Predictive Control based on Dual Decomposition and Quadratic Approximation

Vassilios Yfantis[1], Nigora Gafur[1], Achim Wagner[2] and Martin Ruskowski[1,2]

Abstract— **This paper presents a dual decomposition-based distributed optimization algorithm and applies it to distributed model predictive control (DMPC) problems. The considered DMPC problems are coupled through shared limited resources. Lagrangian duality can be used to decompose an MPC problem, so that each subsystem can compute its individual resource utilization, without sharing information, such as dynamics or constraints, with the other subsystems. The feasibility of the central problem is ensured by the coordination of the subproblems through dual variables which can be interpreted as prices on the shared limited resources. The proposed coordination algorithm makes efficient use of information collected from previous iterations by performing a quadratic approximation of the dual function of the central MPC problem. Aggressive update steps of the dual variables are prevented through a covariance-based step size constraint. The nonsmoothness encountered in dual optimization problems is addressed by the construction of cutting planes, similar to bundle methods for nonsmooth optimization. The cutting planes ensure that the updated dual variables do not lie outside the range of validity of the dual approximation. The proposed algorithm is evaluated on a two-tank system and compared to the standard subgradient method. The results show that the rate of convergence towards the centralized solution can be significantly improved while still preserving privacy between the subsystems through limited information exchange.**

I. Introduction and Related Work

The rise in computing power and the improvements of available optimization algorithms has made optimization-based predictive control a viable option in recent years. Model predictive control (MPC) offers a framework in which a system's performance can be optimized while simultaneously accounting for constraints on the states and inputs [1]. However, the solution of large scale MPC problems can still be challenging. On the one hand, the underlying optimization problem might become too complex to be solved within reasonable computation times [2]. On the other hand, the MPC problem might involve several autonomous subsystems, albeit coupled through constraints, that are not able or willing to exchange information between each other. Examples of the latter instance include the control of power systems, which are coupled through their energy consumption [3]–[5] or reactors belonging to different production units or sites, which are coupled through streams of materials and energy [6]. Distributed MPC (DMPC) addresses the aforementioned issues by splitting the central MPC problem into smaller subproblems [7]. This paper focuses on DMPC using dual decomposition, where coupling constraints between the subproblems are relaxed through the introduction of dual variables [8]. These dual variables can be interpreted as prices for shared resources [3], [9]. The feasibility of the solution is then ensured through the hierarchical coordination of the subproblems through the dual variables. This hierarchical communication structure makes dual decomposition suitable for problems where only limited information can be shared between the subsystems. In [3] dual decomposition-based DMPC is applied to an energy network, where the energy consumption of sub-networks, governed by independent balancing responsible parties (BRPs), is coordinated through dual variables without sharing private information. Similarly, dual decomposition-based DMPC is applied in [4] to optimize power flows in a network of interconnected microgrids while preserving privacy. In [5] both primal and dual decomposition are used for energy management in smart districts using DMPC. In the case of dual decomposition the subsystems are coordinated through prices on the shared resources. In contrast, primal decomposition coordinates the subsystems by directly allocating parts of the shared resources to the involved subproblems. In this way feasibility is ensured in every iteration of the coordination algorithm, whereas dual decomposition only ensures feasibility upon convergence. However, primal decomposition requires far more degrees of freedom on the coordinator level, compared to dual decomposition. In [6] different dual decomposition-based distributed optimization algorithms were used to control semi-batch reactors with coupled feed flows and their performance was compared. Dual decomposition-based distributed optimization algorithms generally exhibit a slow rate of convergence [9], which can render their application for online control problems challenging. This limitation can be overcome, e.g., by accelerated gradient methods were information from previous iterations is exploited in the form of momentum terms [10]. Alternatively, the dual optimization algorithm can be terminated prematurely within a satisfactory range of the optimum. Inexact dual optimization with premature termination in the context of dual decomposition-based DMPC was investigated in [11]. Similarly, premature termination of an algorithm based on augmented Lagrangians for DMPC was studied in [12]. Nevertheless, dual decomposition-based DMPC is still suitable for coupled systems exhibiting slow dynamics and large sampling times, like power systems or chemical processing plants. If real-time computations are required, e.g., in the case of DMPC for cooperative robotic manipulators [13],

[1]The author is with the Chair of Machine Tools and Control Systems, Department of Mechanical and Process Engineering, Technische Universität Kaiserslautern, Kaiserslautern D-67663, Germany (e-mail: vassilios.yfantis@mv.uni-kl.de

[2] The author is with the German Research Center for Artificial Intelligence (DFKI), Kaiserslautern D-67663, Germany

different DMPC approaches can provide better results. An overview of different DMPC approaches and architectures can be found in [7] and [14].

In this work a novel dual decomposition-based distributed optimization algorithm is presented, which updates the dual variables based on a quadratic approximation of the dual function. The remainder of this paper is organized as follows: Section II introduces the considered class of central MPC problems and their decomposition through the introduction of dual variables. Furthermore, the subgradient method for the solution of the dual problem is described. Section III presents the proposed algorithm based on quadratic approximation. The algorithm is demonstrated on an illustrative case study of a two-tank system and compared to the subgradient method in Section IV. The paper is concluded and direction towards future research is provided in Section V.

II. Dual Decomposition-based Distributed MPC

In this section the considered class of MPC problems and their decomposition through Lagrangian duality are introduced. Furthermore, the subgradient method, which can be used to solve the central MPC problem in a distributed fashion, is discussed.

A. Considered class of MPC problems

In this paper MPC problems where N_s subsystems are coupled through constraints on shared resources are considered. The central MPC problem can be formulated as follows:

$$\min_{\mathbf{x}^{0:N_p},\mathbf{u}^{0:N_p-1}} \sum_{i=1}^{N_s}\left[J_i^f(\mathbf{x}_i^{N_p})+\sum_{k=0}^{N_p-1}J_i(\mathbf{x}_i^k,\mathbf{u}_i^k)\right], \quad (1a)$$

$$\text{s.t. } \mathbf{x}_i^{k+1}=\mathbf{A}_i\mathbf{x}_i^k+\mathbf{B}_i\mathbf{u}_i^k,$$
$$\forall i=1,\dots,N_s\ k=0,\dots,N_p-1, \quad (1b)$$
$$\mathbf{x}_i^0=\mathbf{x}_{i,0},\ \forall i=1,\dots,N_s, \quad (1c)$$
$$\mathbf{x}_i^k\in\mathcal{X}_i,\ \forall i=1,\dots,N_s,$$
$$k=0,\dots,N_p, \quad (1d)$$
$$\mathbf{u}_i^k\in\mathcal{U}_i,\ \forall i=1,\dots,N_s,$$
$$k=0,\dots,N_p-1, \quad (1e)$$
$$\sum_{i=1}^{N_s}\mathbf{u}_i^k\le\mathbf{u}_{\max}^k,\ \forall k=0,\dots,N_p-1. \quad (1f)$$

Each subsystem i consists of its individual states $\mathbf{x}_i\in\mathbb{R}^{n_{\mathbf{x}_i}}$ and inputs $\mathbf{u}_i\in\mathbb{R}^{n_\mathbf{u}}$, governed by linear dynamics (1b), with $\mathbf{A}_i\in\mathbb{R}^{n_{\mathbf{x}_i}\times n_{\mathbf{x}_i}}$ and $\mathbf{B}_i\in\mathbb{R}^{n_{\mathbf{x}_i}\times n_\mathbf{u}}$. In the following $\mathbf{x}=(\mathbf{x}_1^T,\dots,\mathbf{x}_{N_s}^T)^T$ and $\mathbf{u}=(\mathbf{u}_1^T,\dots,\mathbf{u}_{N_s}^T)^T$ denote the set of all states and inputs of the central MPC problem respectively. Constraint (1c) corresponds to the states' initial conditions. Each subsystem possesses an individual convex objective function with the running cost $J_i:\mathbb{R}^{n_{\mathbf{x}_i}}\times\mathbb{R}^{n_\mathbf{u}}\to\mathbb{R}$ and the terminal state cost $J_i^f:\mathbb{R}^{n_{\mathbf{x}_i}}\to\mathbb{R}$. Furthermore the states and inputs are subject to convex constraints $\mathcal{X}_i$ and $\mathcal{U}_i$ respectively (1d,1e). The subsystems are coupled in their inputs through constraint (1f). The coupling can be interpreted as systems sharing limited resources, where $\mathbf{u}_{\max}\in\mathbb{R}^{n_\mathbf{u}}$ describes their maximum availability, i.e., how much of the limited resources can be consumed by all subsystems. The MPC problem is solved over a prediction horizon $k=0,\dots,N_p$ with the sampling time T_s and $\mathbf{x}^k=\mathbf{x}(k\cdot T_s)$ and $\mathbf{x}^{0:N_p}=[\mathbf{x}^0,\dots,\mathbf{x}^{N_p}]$.

B. Dual decomposition of the central MPC problem

As described in Section I, a solution of the central MPC problem (1) is not always possible, e.g., due to the problem size or due to privacy concerns between the subsystems. If only a limited amount of information can be shared between the subsystems, the central problem can be decomposed and solved in a distributed fashion. In this work Lagrangian duality is employed to decouple the subsystems as it enables a distributed optimization of problem (1) while only requiring limited information exchange between the subsystems, thus ensuring confidentiality. The Lagrangian of problem (1) is

$$\mathcal{L}(\mathbf{x}^{0:N_p},\mathbf{u}^{0:N_p-1},\boldsymbol{\lambda})=\sum_{i=1}^{N_s}\left[J_i^f(\mathbf{x}_i^{N_p})+\sum_{k=0}^{N_p-1}\left[J_i(\mathbf{x}_i^k,\mathbf{u}_i^k)+\boldsymbol{\lambda}^{k,T}\mathbf{u}_i^k\right]\right]-\sum_{k=0}^{N_p-1}\boldsymbol{\lambda}^{k,T}\mathbf{u}_{\max}^k, \quad (2)$$

where $\boldsymbol{\lambda}=(\boldsymbol{\lambda}^{0,T},\dots,\boldsymbol{\lambda}^{N_p-1,T})^T\in\mathbb{R}^{(N_p-1)\cdot n_\mathbf{u}}$ denotes the dual variables or Lagrange multipliers. The Lagrangian (2) can be decomposed into the Lagrangians of the individual subsystems,

$$\mathcal{L}(\mathbf{x}^{0:N_p},\mathbf{u}^{0:N_p-1},\boldsymbol{\lambda})=\sum_{i=1}^{N_s}\mathcal{L}_i(\mathbf{x}_i^{0:N_p},\mathbf{u}_i^{0:N_p-1},\boldsymbol{\lambda})-\sum_{k=0}^{N_p-1}\boldsymbol{\lambda}^{k,T}\mathbf{u}_{\max}^k, \quad (3)$$

By relaxing the coupling constraints (1e) of the central problem through the dual variables, the MPC problems of the individual subsystems can be solved in a distributed fashion as

$$\min_{\mathbf{x}_i^{0:N_p},\mathbf{u}_i^{0:N_p-1}} \mathcal{L}_i(\mathbf{x}_i^{0:N_p},\mathbf{u}_i^{0:N_p-1},\boldsymbol{\lambda}), \quad (4a)$$

$$\text{s.t. } \mathbf{x}_i^{k+1}=\mathbf{A}_i\mathbf{x}_i^k+\mathbf{B}_i\mathbf{u}_i^k,$$
$$\forall k=0,\dots,N_p-1, \quad (4b)$$
$$\mathbf{x}_i^0=\mathbf{x}_{i,0}, \quad (4c)$$
$$\mathbf{x}_i^k\in\mathcal{X}_i,\ \forall k=0,\dots,N_p, \quad (4d)$$
$$\mathbf{u}_i^k\in\mathcal{U}_i,\ \forall k=0,\dots,N_p-1. \quad (4e)$$

In order to ensure feasibility of the central problem, the dual variables have to be updated iteratively through a coordination algorithm. The coordinator communicates dual variables to the subsystems, which in turn solve their DMPC problem (4) for the current value of $\boldsymbol{\lambda}$. The coordinator subsequently gathers the responses on the shared resource utilization $\mathbf{u}_i^k$ of the subsystems and updates the dual variables accordingly.

The aggregation of the objective values obtained for a given value of the dual variables $\boldsymbol{\lambda}$ through the solution of

the individual MPC problems (4) corresponds to the value of the dual function of the central MPC problem (1)

$$d(\boldsymbol{\lambda}) = \min_{\mathbf{x}^{0:N_p},\mathbf{u}^{0:N_p-1}} \mathcal{L}(\mathbf{x}^{0:N_p}, \mathbf{u}^{0:N_p-1}, \boldsymbol{\lambda}), \tag{5a}$$

$$\text{s.t. } \mathbf{x}_i^{k+1} = \mathbf{A}_i\mathbf{x}_i^k + \mathbf{B}_i\mathbf{u}_i^k, \quad \forall i = 1,\dots,N_s,\ k = 0,\dots,N_p-1, \tag{5b}$$

$$\mathbf{x}_i^0 = \mathbf{x}_{i,0},\ \forall i = 1,\dots,N_s, \tag{5c}$$

$$\mathbf{x}_i^k \in \mathcal{X}_i,\ \forall i = 1,\dots,N_s,\ k = 0,\dots,N_p, \tag{5d}$$

$$\mathbf{u}_i^k \in \mathcal{U}_i,\ \forall i = 1,\dots,N_s,\ k = 0,\dots,N_p-1. \tag{5e}$$

The maximization of the dual function

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}^{(N_p-1)\cdot n_{\mathbf{u}}}} d(\boldsymbol{\lambda}), \tag{6a}$$

$$\text{s.t. } \boldsymbol{\lambda} \geq 0 \tag{6b}$$

is referred to as the dual problem, while the central MPC problem (1) is the corresponding primal problem. Constraints (6b) follows from the Karush-Kuhn-Tucker conditions, as the relaxed coupling constraints (1e) are posed as inequalities [15]. If the primal problem is convex and a feasible primal-dual solution $(\mathbf{x}^{0:N_p*}, \mathbf{u}^{0:N_p-1*}, \boldsymbol{\lambda}^*)$ exists, then strong duality holds

$$d(\boldsymbol{\lambda}^*) = \sum_{i=1}^{N_s} \left[J_i^f(\mathbf{x}_i^{N_p*}) + \sum_{k=0}^{N_p-1} J_i(\mathbf{x}_i^{k*}, \mathbf{u}_i^{k*}) \right], \tag{7}$$

i.e., solving the dual problem is equivalent to solving the primal problem. The dual function is always concave, regardless whether or not the primal problem is convex, i.e., the dual problem (6) is always a convex optimization problem [15]. The basis of dual decomposition-based distributed optimization algorithms is that the dual function (5) can be evaluated in a distributed fashion by solving the individual DMPC problems (4) for a given set of dual variables $\boldsymbol{\lambda}$. In the following subsection the subgradient method for the solution of the dual problem is discussed.

C. Subgradient method

Solving the dual problem (6) through a gradient ascent method is generally not possible, as a gradient of the dual function can not be computed in a distributed fashion. Furthermore, a gradient might not even exist for all values of $\boldsymbol{\lambda}$, as the dual function $d(\boldsymbol{\lambda})$ is generally nonsmooth. However, a subgradient can be computed instead of a gradient. A vector $\boldsymbol{\xi} \in \mathbb{R}^{n_{\boldsymbol{\chi}}}$ is called a subgradient of a convex function $\phi : \mathbb{R}^{n_{\boldsymbol{\chi}}} \to \mathbb{R}$ at a point $\boldsymbol{\chi}_0 \in \mathbb{R}^{n_{\boldsymbol{\chi}}}$, if

$$\phi(\boldsymbol{\chi}) \geq \phi(\boldsymbol{\chi}_0) + \boldsymbol{\xi}^T(\boldsymbol{\chi} - \boldsymbol{\chi}_0),\ \forall \boldsymbol{\chi} \in \text{dom } \phi, \tag{8}$$

holds [16]. The set of all subgradients of a function $\phi(\boldsymbol{\chi})$ at the point $\boldsymbol{\chi}_0$ comprise the subdifferential $\partial\phi(\boldsymbol{\chi}_0)$. In the case of the dual function $d(\boldsymbol{\lambda})$, a subgradient $\mathbf{g}(\boldsymbol{\lambda})$ of the negative dual can be computed through the evaluation of the coupling constraints (1e) for the optimal primal variables $\mathbf{u}^{0:N_p-1*}(\boldsymbol{\lambda})$ obtained for the dual variables $\boldsymbol{\lambda}$ [17], i.e.,

$$\underbrace{\left(\sum_{i=1}^{N_s} \mathbf{u}_i^{k*}(\boldsymbol{\lambda}) - \mathbf{u}_{\max}^k \right)_{\{k=0,\dots,N_p-1\}}}_{=\mathbf{g}(\boldsymbol{\lambda})\in\mathbb{R}^{n_{\boldsymbol{\lambda}}}} \in \partial(-d(\boldsymbol{\lambda})). \tag{9}$$

The subgradient method iteratively updates the dual variables by performing a step in the direction given by the subgradient [18],

$$\boldsymbol{\lambda}^{(c+1)} = \left[\boldsymbol{\lambda}^{(c)} + \alpha^{(c)} \cdot \mathbf{g}(\boldsymbol{\lambda}^{(c)}) \right]^+, \tag{10}$$

where c denotes the iteration index, $\alpha^{(c)} > 0$ is a step size parameter and $[\cdot]^+$ denotes the projection onto the positive orthant. The updated dual variables are subsequently communicated to the subsystems, where the individual DMPC problems (4) are solved in a distributed fashion. In this work the algorithm is terminated when the primal residual,

$$w_p^{(c)} = \| \max\{0, \mathbf{g}(\boldsymbol{\lambda}^{(c)})\} \|_2, \tag{11}$$

the dual residual

$$w_d^{(c)} = \| \boldsymbol{\lambda}^{(c+1)} - \boldsymbol{\lambda}^{(c)} \|_2, \tag{12}$$

and the complementary slackness

$$w_s^{(c)} = \| \boldsymbol{\lambda}^{(c),T} \mathbf{g}(\boldsymbol{\lambda}^{(c)}) \|_2 \tag{13}$$

lie below a defined threshold ϵ. Eq. (11) indicates primal feasibility, eq. (12) indicates convergence of the dual variables and eq. (13) indicates the satisfaction of the complementarity conditions [15]. The subgradient method converges, if the step size is adequately chosen. It must be set large enough to ensure fast convergence, but not too large so as to prevent oscillations or even divergence. However, the algorithm generally exhibits a slow rate of convergence. In the following section a dual decomposition-based distributed optimization algorithm which makes efficient use of previously collected information to speed up the rate of convergence is presented.

III. Quadratically Approximated Dual Ascent

Wenzel et al. [9] proposed to approximate the squared primal residual $w_p^2(\boldsymbol{\lambda})$ through a quadratic function by collecting information obtained in previous iterations and to then update the dual variables through a minimization of this quadratic function. This approach suffers from the drawback that the approximated function is nonconvex and nonsmooth if the subsystems contain changing sets of active local constraints. In contrast, the algorithm proposed in this paper approximates the dual function $d(\boldsymbol{\lambda})$ which might be nonsmooth, but is always concave. Since a smooth approximation of a nonsmooth function is prone to inaccuracies, the update of the dual variables includes constraints that take the nonsmoothness into account, which are not applicable in the algorithm proposed in [9]. The different elements of the algorithm are presented in the following subsections.

A. Quadratic approximation of the dual function

The key of the proposed algorithm is the approximation of the dual function $d(\boldsymbol{\lambda})$ by a quadratic function

$$d_Q^{(c)}(\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{\lambda}^T \mathbf{Q}^{(c)} \boldsymbol{\lambda} + \mathbf{q}^{(c),T} \boldsymbol{\lambda} + q_0^{(c)} \tag{14}$$

in each iteration. The parameters of the quadratic function $\mathbf{Q}^{(c)} \in \mathbb{R}^{n_{\boldsymbol{\lambda}} \times n_{\boldsymbol{\lambda}}}$, $\mathbf{q}^{(c)} \in \mathbb{R}^{n_{\boldsymbol{\lambda}}}$ and $q_0^{(c)} \in \mathbb{R}$ can be obtained through the solution of a regression problem

$$\min_{\mathbf{Q}^{(c)}, \mathbf{q}^{(c)}, q_0^{(c)}} \sum_{\forall j \in \mathcal{J}^{(c)}} \| d_Q^{(c)}(\boldsymbol{\lambda}^{(j)}) - d(\boldsymbol{\lambda}^{(j)}) \|_2^2, \tag{15}$$

where $\mathcal{J}^{(c)} \subseteq \{1, \ldots, c\}$ denotes the dataset used for the regression. In order to perform a regression at least

$$m_{\text{reg,min}} = (n_{\boldsymbol{\lambda}} + 1)(n_{\boldsymbol{\lambda}} + 2)/2 \tag{16}$$

data points have to be collected [9]. To this end, the proposed algorithm initially performs subgradient steps (10) and stores the data

$$\mathcal{B}^{(c)} = \{(\boldsymbol{\lambda}^{(j)}, d(\boldsymbol{\lambda}^{(j)}), \mathbf{g}(\boldsymbol{\lambda}^{(j)})) \in \mathbb{R}^{n_{\boldsymbol{\lambda}}} \times \mathbb{R} \times \mathbb{R}^{n_{\boldsymbol{\lambda}}}, \; 1 \leq j \leq c\}. \tag{17}$$

The collected data includes the dual variables as well as the value of the dual function and a subgradient. Once enough data has been collected, the subset of the data points $\mathcal{J}^{(c)}$ used for the regression problem (15) is selected. In this work the nearest axis point separation (NAPS) algorithm, described in [9], is employed for the data selection. The NAPS algorithm does not select data older than a specified age τ (in terms of past iterations) and aims at selecting points $\boldsymbol{\lambda}^{(j)}$ that are evenly spread around the current value $\boldsymbol{\lambda}^{(c)}$. All points within a specified distance $r_{\boldsymbol{\lambda}}$ are selected. Afterwards, the algorithm cycles through the remaining data points, until enough points for a regression are selected. For more details the reader is referred to [9].

B. Covariance-based step size constraint

In order to prevent too aggressive update steps, which might lead to divergence, Wenzel et al. [9] proposed the use of a step size constraint, based on the covariance matrix of the dual variables used for the approximation. If the used dual variables are summarized in a matrix $\boldsymbol{\Lambda}^{(c)}$, its covariance matrix can be computed and subsequently decomposed through a singular value decomposition,

$$\mathbf{C}^{(c)} = \text{cov}(\boldsymbol{\Lambda}^{(c)}) = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \; \boldsymbol{\Sigma} = \text{diag}(\sigma_i), \tag{18}$$

where $\sigma_i, \; i = 1, \ldots, n_{\boldsymbol{\lambda}}$ denotes the singular values. By scaling the singular values according to

$$\hat{\sigma}_i := \max\{\underline{s}_i, \min\{\sigma_i, \overline{s}_i\}\}, \tag{19}$$

with user defined lower and upper bounds $\underline{s}_i$ and $\overline{s}_i$, a scaled covariance matrix can be computed,

$$\hat{\mathbf{C}}^{(c)} = \mathbf{U}\hat{\boldsymbol{\Sigma}}\mathbf{V}^T, \; \hat{\boldsymbol{\Sigma}} = \text{diag}(\hat{\sigma}_i). \tag{20}$$

The update of the dual variables is then constrained to lie within the ellipsoid

$$\mathcal{E}(\boldsymbol{\Lambda}^{(c)}) = \{\boldsymbol{\lambda} \in \mathbb{R}^{n_{\boldsymbol{\lambda}}} | \; (\boldsymbol{\lambda} - \boldsymbol{\lambda}^{(c)})^T \hat{\mathbf{C}}^{-1} (\boldsymbol{\lambda} - \boldsymbol{\lambda}^{(c)}) \leq \gamma^{(c),2}\}. \tag{21}$$

The parameter γ can be updated according to

$$\gamma^{(c)} = \max\{\log w_p^{(c)}, \underline{\gamma}\}, \tag{22}$$

with a user defined lower bound $\underline{\gamma}$, in order to allow larger steps in the first iterations and smaller steps close to the optimum [9].

C. Bundle cuts

The dual function is usually not smooth, as the set of active constraints of the subsystems might change for changing values of the dual variables. Bundle methods have been shown to be very efficient for nonsmooth optimization problems [16]. Their key idea is to use the previously collected data $\mathcal{B}^{(c)}$ to construct a piece-wise linear over-approximator of the dual function. The dual variables can then be updated by computing an ascent direction of this approximation. In this paper the same collected information is used to further constrain the update of the dual variables. As $\mathbf{g}(\boldsymbol{\lambda}^{(j)})$ is a subgradient of $-d(\boldsymbol{\lambda})$ at the point $\boldsymbol{\lambda}^{(j)}$, the same should hold for the approximated dual function, i.e.,

$$d_Q^{(c)}(\boldsymbol{\lambda}) \leq d(\boldsymbol{\lambda}^{(j)}) + \mathbf{g}^T(\boldsymbol{\lambda}^{(j)})(\boldsymbol{\lambda} - \boldsymbol{\lambda}^{(j)}), \; \forall j \in \mathcal{J}^{(c)}. \tag{23}$$

If condition (23) is not satisfied for the updated dual variables $\boldsymbol{\lambda}^{(c+1)}$, the algorithm has left the range of validity of the approximation. Hence, the updated dual variables are constrained to satisfy the constraints (23) defined by the cutting planes, referred to as bundle cuts in the following.

D. Update problem

To summarize, the proposed algorithm initially performs subgradient update steps (10) until at least $m_{\text{reg,min}}$ data points have been collected. Afterwards, in each iteration regression data is selected and the parameters of the quadratic model are computed through the solution of the regression problem (15). Note that the regression problem can be solved through the inversion of the corresponding Vandermonde matrix. The dual variables are then updated by solving the constrained optimiaztion problem

$$\max_{\boldsymbol{\lambda} \in \mathbb{R}^{n_{\boldsymbol{\lambda}}}} \; d_Q^{(c)}(\boldsymbol{\lambda}), \tag{24a}$$

$$\text{s.t. } \boldsymbol{\lambda} \in \mathcal{E}(\boldsymbol{\Lambda}^{(c)}), \tag{24b}$$

$$d_Q^{(c)}(\boldsymbol{\lambda}) \leq d(\boldsymbol{\lambda}^{(j)}) + \mathbf{g}^T(\boldsymbol{\lambda}^{(j)})(\boldsymbol{\lambda} - \boldsymbol{\lambda}^{(j)}), \; \forall j \in \mathcal{J}^{(c)}, \tag{24c}$$

$$\boldsymbol{\lambda} \geq 0, \tag{24d}$$

instead of subgradient update steps. Constraint (24b) is the covariance-based step size constraint while constraints (24c) are the bundle cuts constraints. Constraint (24d) ensures dual feasibility of the updated variables. Note that since the dual function $d(\boldsymbol{\lambda})$ is always concave the approximated dual function will usually also be concave if suitable regression

data is chosen. Therefore the update problem (24) is a convex quadratically constrained quadratic program, which can be solved efficiently. The quadratic approximation is used to update the dual variables in an ascent direction of the approximated dual function, hence the algorithm describes a quadratically approximated dual ascent (QADA).

IV. Numerical Results

In this section the proposed QADA algorithm is demonstrated on an illustrative case study. A two-tank system is linearized and the levels of the tanks are controlled in a distributed fashion, while respecting constraints on the shared resources.

A. Two-tank system

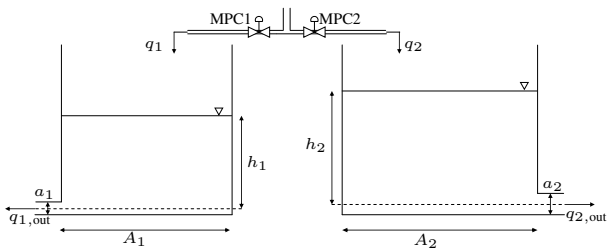


Fig. 1: Two tank system.

The system consisting of two tanks is depicted in Fig. 1. Each tank represents a subsystem, where the level of each tank is controlled in a distributed fashion. Assuming incompressibility of the fluid, the dynamics of the level h_i of each tank i can be described through eq. (25),

$$\begin{aligned}\dot{h}(t) &= -\frac{1}{A_i}\left(q_{i,\text{out}}(t) - q_i(t)\right) \\ &= -\frac{a_i}{A_i}\sqrt{2gh_i(t)} + \frac{1}{A_i}q_i(t),\end{aligned} \tag{25}$$

where A_i and a_i denote the cross-sectional areas of the tank and the outlet respectively, g is the gravitational acceleration and $q_i(t)$ is the flowrate serving as a controlled input of the system. The tanks are decoupled in their dynamics. However, the inputs are limited by a shared resource, i.e.,

$$q_1(t) + q_2(t) \leq q_{\max}(t). \tag{26}$$

The levels of the two tanks are controlled in a distributed fashion without exchanging information such as dynamics, state trajectories or constraints. To this end, dual decomposition is applied to decouple the two MPC problems, as described in Section II.

B. System and algorithm parametrization

For the purpose of the MPC problems, the nonlinear dynamics (25) are linearized around a steady-state, given by the initial conditions $h_{1,0} = 3$ m and $h_{2,0} = 5$ m. The levels of the tanks are constrained by upper bounds $h_1^{\text{UB}} = 4.5$ m and $h_2^{\text{UB}} = 5.5$ m, respectively. The flowrates are constrained by $q_1^{\text{UB}} = 0.45$ m^3/min and $q_2^{\text{UB}} = 0.69$ m^3/min. For the tanks $A_1 = 5$ m^2, $a_1 = 5.27 \cdot 10^{-4}$ m^2, $A_2 = 7$ m^2, and $a_2 = 7.38 \cdot 10^{-4}$ m^2 holds.

Each MPC controller optimizes a convex objective function

$$J_i^f(\mathbf{x}_i^{N_p}) \quad = \|\mathbf{x}_i^{N_p} - \mathbf{x}_i^{N_p,\text{ref}}\|^2_{\mathbf{H}_{\mathbf{x},i}}, \tag{27a}$$

$$J_i(\mathbf{x}_i^k, \mathbf{u}_i^k) \quad = \|\mathbf{x}_i^k - \mathbf{x}_i^{k,\text{ref}}(k)\|^2_{\mathbf{H}_{x,i}} + \|\mathbf{u}_i^k\|^2_{\mathbf{R}_{\mathbf{u},i}}, \tag{27b}$$

where $\|\mathbf{y}\|^2_{\mathbf{P}} = \mathbf{y}^T\mathbf{P}\mathbf{y}$ denotes the weighted squared 2-norm, with $\mathbf{H}_{\mathbf{x},1} = \mathbf{H}_{\mathbf{x},2} = 1$, $\mathbf{R}_{\mathbf{u},1} = 0.1$ and $\mathbf{R}_{\mathbf{u},2} = 0.02$. The states $\mathbf{x}_i$ of the subsystems correspond to the levels h_i, while the flowrates q_i serve as control inputs $\mathbf{u}_i$. The system exhibits slow dynamics, so that dual decomposition-based DMPC is applicable. The sampling time is set to $T_s = 1$ min and the prediction horizon to $N_p = 10$.

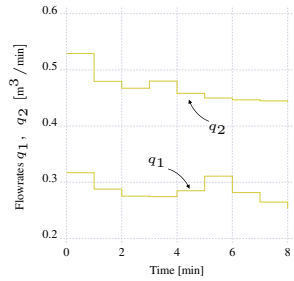
The dual decomposition algorithms are initialized with $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$, i.e., with a decentral solution. The step size parameter of the subgradient method is adapted according to

$$\alpha^{(c)} = \alpha^{(0)}/\sqrt{c}, \tag{28}$$

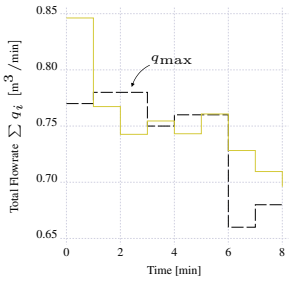
with $\alpha^{(0)} = 0.1$. For QADA, subgradient steps are performed until $m_{\text{reg,min}}$ data points have been collected. The parameters of the point selection algorithm are set to $r_{\boldsymbol{\lambda}} = 5 \cdot 10^{-5}$ and $\tau = 2 \cdot m_{\text{reg,min}}$. The ellipsoid axes of the step size constraint (24b) are scaled by $\underline{s} = 2 \cdot 10^{-5}$, $\overline{s} = 10^{-3}$ and $\underline{\gamma} = 1$. The maximum number of iterations is set to $k_{\max} = 500$ and the convergence tolerances to $\epsilon = 10^{-3}$. The parameters where chosen such that fast convergence is achieved without too aggressive update steps, which would lead to oscillations or divergence. Both the DMPC problems and the QADA update problem (24) are implemented in *Matlab* using *CasADi* [19]. The interior point solver *IPOPT* [20] is applied to solve the optimization problems.

C. Simulation results

The results for the dual decomposition-based DMPC of the two-tank system are depicted in Fig. 2. In Fig. 2a the two MPC controllers compute their respective flowrates in a decentral fashion (for $\boldsymbol{\lambda} = \mathbf{0}$) without accounting for the limitation of the shared resource and for the resource utilization computed by the other controller. This approach leads to an overutilization of the shared resource, as seen in Fig. 2b. In order to achieve feasibility and convergence towards the solution of the central MPC problem, the subgradient method and QADA are applied to coordinate the resource consumption and their performance is compared. Fig. 2c illustrates the evolution of the primal (11) and dual (12) residuals for the two algorithms. The subgradient method requires 201 iterations to converge to the solution of the central problem. In comparison, QADA initially is identical to the subgradient method while data points are being collected through subgradient update steps. Once enough points are collected, the dual variables are instead updated through problem (24). While the algorithm initially searches in the wrong direction for the first step, indicated by the increase of the residuals, it quickly converges to the solution of the central problem within 82 iterations. Fig. 2d depicts the flowrates computed by the DMPC controllers upon convergence of the QADA algorithm. As seen in Fig. 2e, the

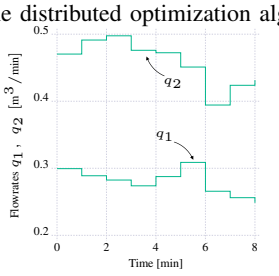


(a) Flowrates of decentral controllers.

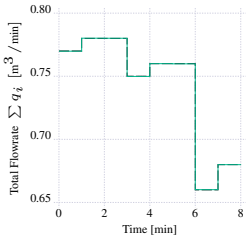


(b) Total shared resource utilization of decentral controllers.

(c) Evolution of the norm of the primal and dual residuals for the distributed optimization algorithms.



(d) Flowrates of the controllers upon convergence of the dual maximization problem.



(e) Total shared resource utilization upon convergence of the dual maximization problem.

Fig. 2: Results for the distributed control of the considered two tank system.

resource constraints are satisfied over the entire prediction horizon. The results upon convergence of the subgradient method are almost identical and are therefore omitted at this point. Nevertheless, Fig. 2c shows that QADA exhibits a smaller value of the primal residual upon convergence. This can be interpreted as a termination with a smaller violation of the constraints on the shared resources.

V. Conclusion and Outlook

This paper presented a novel dual decomposition-based distributed optimization algorithm. The algorithm relies on a quadratic approximation of the dual function and updates the dual variables by solving a constrained optimization problem in each iteration. The algorithm can be used to solve distributed MPC problems while preserving privacy between the involved subsystems. The approach was evaluated on an illustrative case study consisting of two tanks with shared resources. The proposed algorithm showed superior performance in terms of the required number of iterations compared to the standard subgradient method. Future work will focus on the comparison of the proposed algorithm to other state-of-the-art dual decomposition-based distributed optimization methods, e.g., the alternating direction method of multipliers (ADMM) or bundle methods [16]. Furthermore, the QADA algorithm can be improved by employing a more efficient algorithm in the initial sampling stage.

References

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, Vol. 36, no. 6, pp. 789–814, 2000.

[2] P. Pflaum, M. Alamir, and M. Y. Lamoudi, "Scalability study for a hierarchical NMPC scheme for resource sharing problems," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 1468–1473.

[3] B. Biegel, P. Andersen, J. Stoustrup, and J. Bendtsen, "Congestion management in a smart grid via shadow prices," *IFAC Proceedings Volumes*, Vol. 45, no. 21, pp. 518–523, 2012.

[4] M. Razzanelli, E. Crisostomi, L. Pallottino, and G. Pannocchia, "Distributed model predictive control for energy management in a network of microgrids using the dual decomposition method," *Optimal Control Applications and Methods*, Vol. 41, no. 1, pp. 25–41, 2020.

[5] P. Pflaum, M. Alamir, and M. Y. Lamoudi, "Comparison of a primal and a dual decomposition for distributed MPC in smart districts," in *2014 IEEE international conference on smart grid communications (SmartGridComm)*. IEEE, 2014, pp. 55–60.

[6] L. S. Maxeiner and S. Engell, "Comparison of dual based optimization methods for distributed trajectory optimization of coupled semi-batch processes," *Optimization and Engineering*, Vol. 21, no. 3, pp. 761–802, 2020.

[7] P. D. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers & Chemical Engineering*, Vol. 51, pp. 21–41, 2013.

[8] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, no. 11 (3), pp. 399–417, 1963.

[9] S. Wenzel, F. Riedl, and S. Engell, "An efficient hierarchical market-like coordination algorithm for coupled production systems based on quadratic approximation," *Computers & Chemical Engineering*, Vol. 134, p. 106704, 2020.

[10] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, Vol. 49, no. 3, pp. 829–833, 2013.

[11] J. Köhler, M. A. Müller, and F. Allgöwer, "Distributed model predictive control—recursive feasibility under inexact dual optimization," *Automatica*, Vol. 102, pp. 1–9, 2019.

[12] F. Farokhi, I. Shames, and K. H. Johansson, "Distributed MPC via dual decomposition and alternative direction method of multipliers," in *Distributed Model Predictive Control Made Easy*. Springer, 2014, pp. 115–131.

[13] N. Gafur, G. Kanagalingam, and M. Ruskowski, "Dynamic collision avoidance for multiple robotic manipulators based on a non-cooperative multi-agent game," *arXiv preprint arXiv:2103.00583*.

[14] R. Scattolini, "Architectures for distributed and hierarchical model predictive control – A review," *Journal of process control*, Vol. 19, no. 5, pp. 723–731, 2009.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[16] A. Bagirov, N. Karmitsa, and M. M. Mäkelä, *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer, 2014.

[17] B. Biegel, J. Stoustrup, and P. Andersen, "Distributed MPC via dual decomposition," in *Distributed Model Predictive Control Made Easy*. Springer, 2014, pp. 179–192.

[18] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[19] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, Vol. 11, no. 1, pp. 1–36, 2019.

[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, Vol. 106, no. 1, pp. 25–57, 2006.