51st IEEE Conference on Decision and Control
December 10-13, 2012. Maui, Hawaii, USA

Computational Aspects of Distributed Optimization in Model Predictive Control

Christian Conte[1], Tyler Summers[1], Melanie N. Zeilinger[2], Manfred Morari[1] and Colin N. Jones[2]

Abstract— **This paper presents a systematic computational study on the performance of distributed optimization in model predictive control (MPC). We consider networks of dynamically coupled systems, which are subject to input and state constraints. The resulting MPC problem is structured according to the system's dynamics, which makes the problem suitable for distributed optimization. The influence of fundamental aspects of distributed dynamic systems on the performance of two particular distributed optimization methods is systematically analyzed. The methods considered are dual decomposition based on fast gradient updates (DDFG) and the alternating direction method of multipliers (ADMM), while the aspects analyzed are coupling strength, stability, initial state, coupling topology and network size. The methods are found to be sensitive to coupling strength and stability, but relatively insensitive to initial state and topology. Moreover, they scale well with the number of subsystems in the network.**

I. INTRODUCTION

This paper presents a computational study on the use of distributed optimization in MPC. The class of systems for which such a control approach is useful are networks of subsystems which are coupled in either their dynamics, their objectives or their constraints. It is assumed that each subsystem can only take local measurements, while communication among coupled subsystems is possible.

Various distributed MPC schemes for this class of systems have been proposed in the literature. They can roughly be divided into two categories, namely non-iterative and iterative methods. In non-iterative methods, neighboring subsystems communicate once per time step only, while in iterative methods they communicate several times. Among the non-iterative schemes are MPC formulations where the local deviation from a given feasible trajectory is explicitly constrained, e.g. [1]. Furthermore, there are approaches where dynamic coupling is taken into account as disturbance [2], [3]. Thus, non-iterative schemes are usually conservative due to the fact that local control actions are based on a worst-case assumption regarding neighboring subsystems.

In comparison to its non-iterative counterpart, iterative distributed MPC is less conservative since multiple communication steps allow for negotiation among the subsystems. In this way, neighboring subsystems can strive for a consensus on local trajectories which are beneficial or even optimal in the global sense. A standard framework for this negotiation process is given by distributed optimization. Specific distributed optimization methods that have been proposed for distributed MPC are dual decomposition [4], proximal center decomposition [5], primal decomposition [6] and a hybrid Gauss-Jacobi method [7]. From a global system perspective, the principle of iterative distributed MPC is very similar to standard centralized MPC, except for the fact that the MPC problem is solved in a distributed manner.

Despite its prominent position in the literature, the performance of distributed optimization in MPC is poorly understood. It is well known that in some cases distributed optimization methods converge slowly, whereas in contrast e.g. [8] points out that the alternating direction method of multipliers (ADMM), a particular distributed optimization method, often converges to a satisfactory accuracy in a few tens of iterations. The primal objective of this paper is therefore to provide a systematic computational study on the performance of two well-established distributed optimization methods, namely dual decomposition based on fast gradient updates (DDFG) and ADMM, in MPC. In particular, the sensitivity of their performance with respect to a comprehensive but not exhaustive list of aspects fundamental to distributed MPC is investigated: (i) the initial state of the problem, (ii) the coupling topology in the network of subsystems, (iii) the coupling strength among neighboring subsystems, (iv) the stability of the decoupled subsystems and (v) the number of subsystems in the network. To the best of the authors' knowledge, this is the first study of this kind published in the context of distributed MPC.

The paper is organized as follows: In Section II the distributed MPC problem is presented, while in Section III specific distributed optimization methods are introduced. Section IV contains the computational results and the discussions thereof, while Section V concludes the paper.

II. PRELIMINARIES

A. Notation

For a matrix $S \in \mathbb{R}^{n \times m}$, the norm $\|S\|_2$ denotes its largest singular value. For a set of vectors $\{x_i\}_{i \in \mathcal{M}}$, where $x_i \in \mathbb{R}^{n_i}$ and $i \in \mathcal{M} \subseteq \mathbb{N}$, $\mathrm{col}_{i \in \mathcal{M}}(x_i)$ denotes a vector in which all elements of $\{x_i\}_{i \in \mathcal{M}}$ are stacked.

B. Distributed Linear Dynamic Systems

We consider discrete-time linear dynamic systems of the form

$$x^+ = Ax + Bu \ , \qquad (1)$$

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number FP7-ICT-2009-4 248940.

[1] Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland cconte|tsummers|morari@control.ee.ethz.ch

[2] Automatic Control Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland melanie.zeilinger|colin.jones@epfl.ch

978-1-4673-2064-1/12/$31.00 ©2012 IEEE

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^p$. System (1), in the following denoted as the global system, consists of M subsystems, each of which has a state $x_i \in \mathbb{R}^{n_i}$ and an input $u_i \in \mathbb{R}^{p_i}$. We denote the set of subsystem indices as $\mathcal{M} = \{1, \ldots, M\}$. Thus, the global state and input vector can be written as

$$x = \mathrm{col}_{i \in \mathcal{M}}(x_i) \ , \ u = \mathrm{col}_{i \in \mathcal{M}}(u_i) \ . \tag{2}$$

The dynamics of each subsystem are

$$x_i^+ = \sum_{j=1}^{M} (A_{ij} x_j + B_{ij} u_j) \ , \ \forall i \in \mathcal{M} \ , \tag{3}$$

where A_{ij} and B_{ij} are sub blocks of the global matrices A and B. Furthermore, states and inputs are locally constrained as

$$x_i \in \mathcal{X}_i \ , \ u_i \in \mathcal{U}_i \ , \ \forall i \in \mathcal{M} \ . \tag{4}$$

The coupling among the subsystems is described by an underlying graph, which can also be used to define the notion of neighboring subsystems.

Definition II.1 (Coupling graph) *The coupling graph is the directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, whose vertex set $\mathcal{V}$ contains one vertex v_i per subsystem $i \in \mathcal{M}$. The set of edges $\mathcal{E}$ contains a directed edge (v_i, v_j), if $A_{ji} \neq 0 \vee B_{ji} \neq 0$. Subsystem i is a neighbor of subsystem j if $(v_i, v_j) \in \mathcal{E}$. The set of neighbors of subsystem i is denoted as $\mathcal{N}_i$ and the set of subsystems to which i is a neighbor is denoted as $\bar{\mathcal{N}}_i$.*

C. Structured MPC

In this paper, we consider the problem of regulating constrained distributed linear systems by model predictive control (MPC), where a finite horizon constrained optimal control problem is solved online at every time instant. This problem is parametric in the initial state x_0 and the control law is defined as the first element of the resulting input trajectory. The problem's objective function has the form

$$\min_{\mathbf{x}, \mathbf{u}} \ V_f(x(N)) + \sum_{k=1}^{N-1} l(x(k), u(k)) \ , \tag{5}$$

where N denotes the finite horizon of the MPC problem. Furthermore, the vectors $\mathbf{x} = \mathrm{col}_{k \in \{0, \ldots, N\}}(x(k))$ and $\mathbf{u} = \mathrm{col}_{k \in \{0, \ldots, N-1\}}(u(k))$ denote the predicted state and input trajectories, which are subject to constraints (3) and (4). The function $l(x, u) : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}$ is the stage cost and the function $V_f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is the terminal cost, which, in combination with an appropriate terminal constraint $x(N) \in \mathcal{X}_f$, guarantees stability of the closed-loop system [9].

In order for distributed optimization to be meaningful, the MPC problem needs to exhibit the same structure as the dynamic system. Constraints (3) and (4) are naturally structured, while an approach to synthesize a structured cost function and a structured terminal constraint was recently proposed in [10]. Thus, the MPC problem can be written in the equivalent form

$$\min_{y_i \in \mathcal{Y}_i(x_0^i) \forall i \in \mathcal{M}, z} \ \sum_{i=1}^{M} J_i(y_i) \tag{6a}$$

$$\text{s.t. } y_i = E_i z \quad \forall i \in \mathcal{M} \ , \tag{6b}$$

where the vector z contains the predicted state and input trajectories of the global system, i.e.

$$z = \mathrm{col}_{i \in \mathcal{M}}(z_i) \ . \tag{7}$$

Naturally, vector z_i represents the predicted state and input trajectories of subsystem i. Furthermore, for every $i \in \mathcal{M}$, every row of E_i is a unit vector with elements in $\{0, 1\}$. Thus, the coupling graph $\mathcal{G}$ is encoded in the matrices E_i and the vector y_i can be seen as a vector of local copies of those variables in z, which affect subsystem i. This vector of local copies can be segmented further as

$$y_i = \mathrm{col}_{j \in \mathcal{N}_i \cup i}(y_{ij}) = \mathrm{col}_{j \in \mathcal{N}_i \cup i}(E_{ij} z_j) \ , \tag{8}$$

where y_{ij} represents those entries in y_i, which represent variables copied from subsystem $j \in \mathcal{N}_i$ and E_{ij} denotes the respective sub block of E_i. In the same context, the set $\mathcal{Y}_i(x_0^i)$ denotes the local constraints affecting subsystem i. Specifically, these are the initial- and terminal constraint, the dynamics according to (3) and the state and input constraints according to (4). Due to the initial state constraint, $\mathcal{Y}_i(x_0^i)$ is a function of x_0^i, the initial state of subsystem i. Finally, $J_i(y_i)$ denotes the local cost function of subsystem i.

Remark II.2 *The particular formulation of problem* (6) *was chosen due to the fact that it is directly compatible with standard distributed optimization methods (see Section III).*

III. DISTRIBUTED OPTIMIZATION

In this section, distributed optimization methods are introduced which are capable of solving constrained problems such as (6), given they are convex. Most distributed optimization methods for convex problems are of first-order type and rely on a decomposition of the dual problem. In this paper, the performance of DDFG, a method relying on the Lagrangian

$$L(y, z, \lambda) = \sum_{i=1}^{M} J_i(y_i) + \lambda_i^T (y_i - E_i z) \ , \tag{9}$$

is compared to the performance of ADMM, a method relying on the augmented Lagrangian

$$L_\rho(y, z, \lambda) = \sum_{i=1}^{M} J_i(y_i) + \lambda_i^T (y_i - E_i z) + \frac{\rho}{2} \|y_i - E_i z\|_2^2 \ . \tag{10}$$

For a more in-depth overview on distributed optimization, the reader is referred to [8], [11].

A. Dual Decomposition Based on Fast Gradient Updates

The Lagrangian (9) is separable into M terms, each of which depends on the variables of one subsystem only. Hence,

$$L(y, z, \lambda) = \sum_{i=1}^{M} L_i(y_i, z_i, \lambda) \ , \tag{11}$$

where for every subsystem $i \in \mathcal{M}$

$$L_i(y_i, z_i, \lambda) = J_i(y_i) + \lambda_i^T y_i - \sum_{j \in \bar{\mathcal{N}}_i \cup i} \lambda_{ji}^T E_{ji} z_i \tag{12}$$

and where λ_{ji} is the respective sub vector of λ_j. Thus, also the dual function of problem (6) is separable as

$$d(\lambda) = \sum_{i=1}^{M} d_i(\lambda) = \sum_{i=1}^{M} \min_{y_i \in \mathcal{Y}_i(x_0^i), z_i} L_i(y_i, z_i, \lambda) \ . \quad (13)$$

The dual of problem (6), $\max_\lambda d(\lambda)$, can then be solved by the procedure described in Algorithm 1, where L denotes the Lipschitz constant of the gradient $\nabla d(\lambda)$ of the dual function.

Algorithm 1 Dual Decomposition Based on Fast Gradient Updates (DDFG)

```
Input: Lipsch. constant of $\nabla d(\lambda)$ $L$, $\alpha = \frac{\sqrt{5}-1}{2}$, $\lambda = \nu = 0$
$\forall i \in \mathcal{M}$ in parallel:
repeat
    $(y_i^+, z_i^+) = \arg\min_{y_i \in \mathcal{Y}_i(x_0^i), z_i} L_i(y_i, z_i, \nu)$
    communicate $z_i^+$ to all $j \in \mathcal{N}_i$
    $\lambda_i^+ = \nu_i + \frac{1}{L}(y_i^+ - E_i z^+)$
    $\alpha^+ = \frac{\alpha}{2}(\sqrt{\alpha^2 + 4} - \alpha)$
    $\beta = \frac{\alpha(1-\alpha)}{\alpha^2 + \alpha^+}$
    $\nu_i^+ = \lambda_i^+ + \beta(\lambda_i^+ - \lambda_i)$
until convergence
```

Remark III.1 *Algorithm 1 converges to a dual optimizer λ^* if for every $i \in \mathcal{M}$, $J_i(y_i)$ is strictly convex. The primal optimizer $col_{i \in \mathcal{M}}(y_i^*, z_i^*)$ is found as a byproduct [11].*

Algorithm 1 is a distributed version of the dual ascent method, in which the gradient updates are done in an optimal way according to [12]. In [13], this approach has recently been customized for the dual of MPC problems. In particular, the bound

$$q_\epsilon \leq \max\left\{ \left\lceil 2\sqrt{\frac{L}{\epsilon}} \|\lambda^*(x_0)\|_2 - 2 \right\rceil, 0 \right\} \quad (14)$$

is implied, where q_ϵ denotes the number of iterations required to achieve an accuracy $d(\lambda^*) - d(\lambda) \leq \epsilon$ in the dual objective value. Furthermore, $\lambda^*(x_0)$ is a dual optimizer for problem (6). Note that [13] provides an explicit formula to find the Lipschitz constant L, whereas $\|\lambda^*(x_0)\|_2$ can generally only be found by solving the optimization problem.

B. Alternating Direction Method of Multipliers

The alternating direction method of multipliers is a parallelizable version of the method of multipliers, which was originally designed as a robust version of the dual ascent method. While the dual ascent method is only applicable to problems with smooth dual functions, the method of multipliers also handles problems with general concave dual functions [8].

The augmented Lagrangian (10) can be written as

$$L_\rho(y, z, \lambda) = \sum_{i=1}^{M} L_\rho^i(y_i, z, \lambda) \ , \quad (15)$$

where

$$L_\rho^i(y_i, z, \lambda) = J_i(y_i) + \lambda_i^T (y_i - E_i z) + \frac{\rho}{2} \|y_i - E_i z\|_2^2 \ . \quad (16)$$

The dual of problem (6) can then be solved by the procedure described in Algorithm 2.

Algorithm 2 Alternating Direction Method of Multipliers (ADMM)

```
1: $\forall i \in \mathcal{M}$ in parallel:
2: choose initial $\lambda_i = 0$, $z = 0$
3: repeat
4:     $y_i^+ = \arg\min_{y_i \in \mathcal{Y}_i(x_0^i)} L_\rho^i(y_i, z, \lambda)$
5:     communicate $y_i^+$ to all $j \in \mathcal{N}_i$
6:     $z_i^+ = \frac{1}{|\bar{\mathcal{N}}_i + 1|} \sum_{j \in \bar{\mathcal{N}}_i \cup i} E_{ji}^T \left( y_{ji}^+ + \frac{1}{\rho} \lambda_{ji} \right)$
7:     communicate $z_i^+$ to all $j \in \bar{\mathcal{N}}_i$
8:     $\lambda_i^+ = \lambda_i + \rho(y_i^+ - E_i z^+)$
9: until convergence
```

Remark III.2 *In Algorithm 2, the residuals $y_i - E_i z$ converge asymptotically to zero for all $i \in \mathcal{M}$ and the objective value $\sum_{i \in \mathcal{M}} J_i(y_i)$ converges asymptotically to the primal optimum if all functions $J_i(y_i)$ are closed, proper and convex and the unaugmented Lagrangian* (10) *has a saddle point [8].*

Remark III.3 *In the averaging step 6 of Algorithm 2, it is assumed for ease of notation, that every $i \in \mathcal{M}$ copies the complete vector z_j of all neighbors $j \in \mathcal{N}_i$. In practice, this is not always necessary.*

IV. COMPUTATIONAL STUDY

In this section, the performance of the distributed optimization methods introduced in Section III is analyzed. The performance of these methods is measured in number of iterations until a specific stopping criterion is met. Specifically, the influence of some fundamental aspects in distributed MPC on the performance is investigated. The study is set up in the spirit of a sensitivity analysis. Hence, we introduce one specific test case to isolate the impact of each aspect on the performance. In each of these test cases, the parameters that reflect the aspect under consideration are varied, while the remaining parameters are held at constant values.

The interpretation of observed performance trends is often difficult. However, in case of DDFG, the bound in (14) can be used as a performance indicator. In particular, the norm of the dual optimizer $\|\lambda^*(x_0)\|_2$ can serve as a source for interpretation. Since the dual variables represent prices on constraint violations, $\|\lambda^*(x_0)\|_2$ can be viewed as the price of consensus. This price determines on one hand a rigorous mathematical upper bound on the number of iterations to convergence and can on the other hand often be used to intuitively connect problem aspects such as system size or coupling strength with performance results. Therefore, in the presentation of all computational results, the number of iterations to convergence for DDFG is illustrated alongside $\|\lambda^*(x_0)\|_2$.

Remark IV.1 *A second parameter that affects the bound in* (14) *is the Lipschitz constant L of $\nabla d(\lambda)$. According to [13] however, in the problem formulation* (6)*, L depends solely on the Hessian of the primal cost function and the coupling constraints* (6b)*. Hence, L is only determined by the coupling graph and therefore only considered when investigating the impact of the coupling topology on the performance in Section IV-B.*

Throughout the computations, we consider subsystems of the form

$$x_i^+ = A_{ii}x_i + B_i u_i + \sum_{j\in\mathcal{N}_i} A_{ij}x_j \quad \forall i \in \mathcal{M} \ , \qquad (17)$$

where the matrices A_{ij} are upper-triangular matrices in $\mathbb{R}^{2\times 2}$ and $B_i \in \mathbb{R}^2$ where the upper entry is zero. For each subsystem i, the matrices A_{ii} and B_i have equal non-zero entries and the matrices A_{ij} have equal non-zero entries for all $j \in \mathcal{N}_i$. These entries can be used to affect stability and coupling properties of the system. The amount to which subsystem i is coupled is measured as

$$\sigma_c^i = \frac{\|A_{\mathcal{N}_i}\|_2}{\|A_{ii}\|_2} \quad \forall i \in \mathcal{M} \ , \qquad (18)$$

where for every subsystem i, $A_{\mathcal{N}_i}$ is composed of the matrices A_{ij} for all $j \in \mathcal{N}_i$. Furthermore, the individual stability σ_s^i of every decoupled subsystem j is measured by the largest eigenvalue of the matrix A_{ii}. Note that even if all A_{ii} are stable, the global system can still be unstable due to coupling effects.

We compare DDFG and ADMM when applied to problem (6) under system dynamics (17). In all computations, the methods are run for a maximum number of 2000 iterations and the step size parameter in ADMM is $\rho = 120$. As a stopping criterion for both methods we use two combined conditions. The first one is a relative condition on the primal residual

$$\frac{\|\text{col}_{\{1,\dots,M\}}(y_i - E_i z)\|_\infty}{\|z\|_\infty} \leq 10^{-3} \ , \qquad (19)$$

and the second condition is a relative condition on the difference of the primal objective value to the optimum

$$\frac{|\sum_{i=1}^M J_i(y_i) - J_i(y_i^*)|}{|\sum_{i=1}^M J_i(y_i^*)|} \leq 10^{-3} \ . \qquad (20)$$

As a reference primal optimum, the centralized solution provided by the commercial solver CPLEX is used. It is evident that neither condition (19) nor condition (20) can be checked online and in a distributed way. However, the problem of imposing practical distributed stopping criteria is non-trivial and beyond the scope of this paper.

For every individual MPC problem, data sets containing the solutions for a number of randomly chosen initial conditions are generated. The findings are illustrated in box plots, where the upper and lower end of the box denote the 25th and 75th percentile of the data, the median is marked as a dot and the whiskers span the whole range of the data.

MPC parameters throughout the computations (if not mentioned otherwise): Time horizon $N = 10$; local state constraints $\mathcal{X}_i = \{x_i \in \mathbb{R}^2 | \|x_i\|_\infty \leq 10\}$, local input constraints $\mathcal{U}_i = \{u_i \in \mathbb{R}^2 | \|u_i\|_\infty \leq 10\}$; quadratic local stage cost $l_i(x_{\mathcal{N}_i}, u_i) = x_{\mathcal{N}_i}^T Q_i x_{\mathcal{N}_i} + u_i^T R_i u_i$ with $Q_i = I$ and $R_i = I$; quadratic local terminal cost $V_f^i(x_i) = x_i^T P_i x_i$ with $P_i = Q_i$; terminal constraint $\mathcal{X}_f = \mathcal{X}_1 \times \ldots \times \mathcal{X}_M$.

Remark IV.2 *The terminal cost and constraint chosen as above do not guarantee stability of the resulting closed-loop MPC controller. The focus of this section, however, is on computational aspects rather than stability. A way to design stabilizing terminal costs and constraints that fit the framework of this computational study was recently presented in [10].*

A. Coupling Strength and Stability

Scenario: To analyze the impact of the coupling strength, the subsystem dynamics (17) are scaled such that for all $i \in \mathcal{M}$ the value σ_c^i is uniformly varied in $\{0.2, 0.4, \ldots, 2.0\}$, while $\sigma_s^i = 1$. Likewise, to analyze the impact of local stability, the subsystem dynamics are scaled such that for all $i \in \mathcal{M}$ $\sigma_c^i = 1$ and σ_s^i is varied in $\{0.2, 0.4, \ldots, 2.0\}$. For each combination (σ_s^i, σ_c^i), problem (6) is solved for 100 random initial conditions.

Findings: Figure 1 illustrates the number of iterations to convergence for ADMM and DDFG as well as the magnitude of $\|\lambda^*(x_0)\|_2$ for varying σ_c^i, Figure 2 illustrates the same quantities for varying σ_s^i. The number of iterations to convergence increases with both of these quantities.

Interpretation: For either varying σ_c^i or σ_s^i, the number of iterations to convergence mainly for DDFG, but also for ADMM, is correlated with $\|\lambda^*(x_0)\|_2$. This allows the following interpretation:

- *Coupling Strength:* As the coupling strength increases, subsystem i has more incentive to influence the state trajectories of subsystems in $\mathcal{N}_i$. Therefore, the price of consensus increases and so does the number of iterations.
- *Stability:* As system i becomes more unstable, it becomes more sensitive to its neighbors' trajectories. Thus, the incentive to influence these increases and as a consequence, the price of consensus and the number of iterations to convergence is also increased.

B. Coupling Topology

Scenario: For 10 subsystems of form (17), the coupling graph is varied according to four topologies (see Fig. 3): (a) chain, (b) ring, (c) 2-hop ring, (d) 3-hop ring. For each topology, problem (6) is solved for 100 random initial conditions. The coupling and stability parameters are $\sigma_c^i = \sigma_s^i = 1$ for every $i \in \mathcal{M}$.

Findings: Figure 4 illustrates the number of iterations to convergence for ADMM and DDFG as well as the magnitude of $\|\lambda^*(x_0)\|_2$. The number of iterations to convergence increases with the graph connectivity and so does the magnitude of $\|\lambda^*(x_0)\|_2$. The Lipschitz constant L of $d(\lambda)$ is

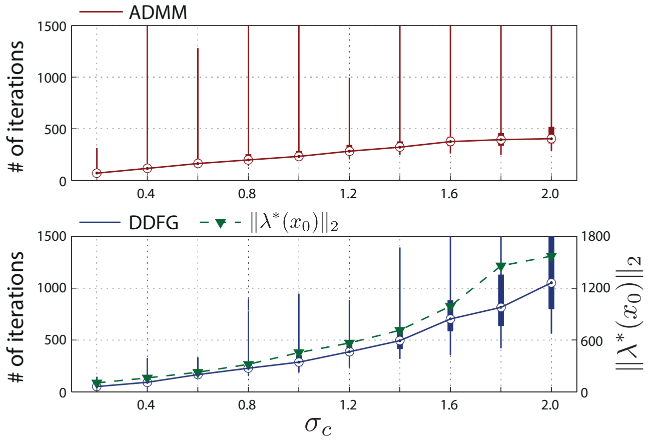


Fig. 1: Number of iterations to convergence under ADMM and DDFG for a network of 10 coupled double-integrators with $\sigma_s^i = 1$ and $\sigma_c^i \in \{0.2, 0.4, \ldots, 2.0\}$. The median of $\|\lambda^*(x_0)\|_2$ is depicted in the lower plot.

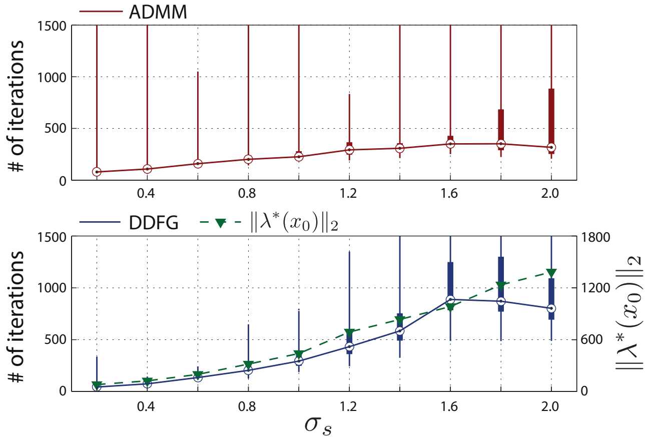
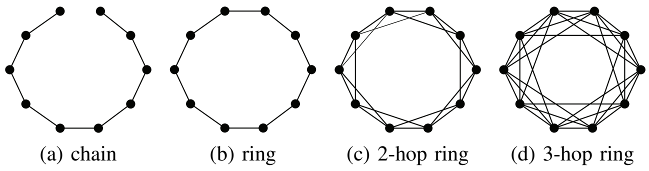


Fig. 2: The analog to Figure 1 with $\sigma_c^i = 1$ and $\sigma_s^i \in \{0.2, 0.4, \ldots, 2.0\}$.

found as 1.5 for chain and ring, 2.5 for the two-hop ring and 3.5 for the three-hop ring.

Interpretation: While, according to the consensus literature, an improvement in performance with increasing graph connectivity would be expected, our results show an increase in the number of iterations. One possible explanation for this behavior is the increase in the dimension of λ^*, which affects the size of $\|\lambda^*(x_0)\|_2$ if the magnitude of the dual optimizer's entries remain comparable. A larger bound (14) then suggests an increase in the number of iterations to convergence. Another possible explanation is the increase in L, which has a similar effect on bound (14).



(a) chain (b) ring (c) 2-hop ring (d) 3-hop ring

Fig. 3: Graph topologies under consideration.

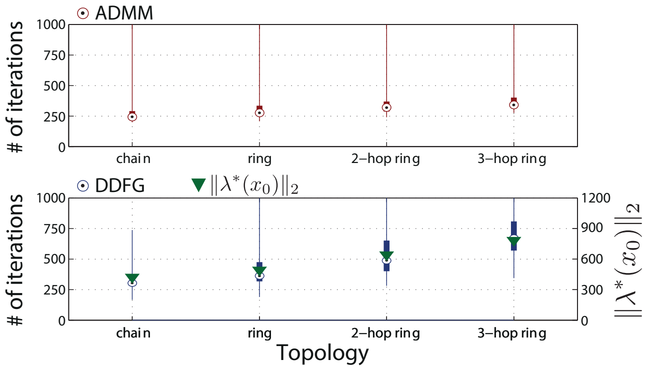


Fig. 4: Number of iteration to convergence for different coupling topologies with ADMM and DDFG, median of the dual optimizer $\|\lambda^*(x_0)\|$.

C. Initial State

Scenario: We consider a chain of 10 subsystems of form (17) with $\sigma_c^i = \sigma_s^i = 1$ for every $i \in \mathcal{M}$. 100 random initial states on the boundary of the region of attraction are generated (by bisection). Subsequently, problem (6) is solved for scalings of these initial conditions, with scalar scaling factors in $\{0.2, 0.4, \ldots, 1.0\}$.

Findings: Figure 5 illustrates an increase in the number of iterations for both DDFG and ADMM with the initial state approaching the boundary of the feasible set. Furthermore, the spread in the number of iterations (visible by the larger span between the 25th percentile and the 75th percentile) becomes larger and the probability of outliers increases.

Interpretation: The number of iterations to convergence mainly for DDFG, but also for ADMM, seems to be correlated with the magnitude of $\|\lambda^*(x_0)\|_2$. This behavior is possibly caused by the fact that as the initial state approaches the boundary of the feasible set, the range of possible control moves is decreased for every subsystem $i \in \mathcal{M}$. Such a decrease is likely to increase the price of consensus and thus also the number of iterations.

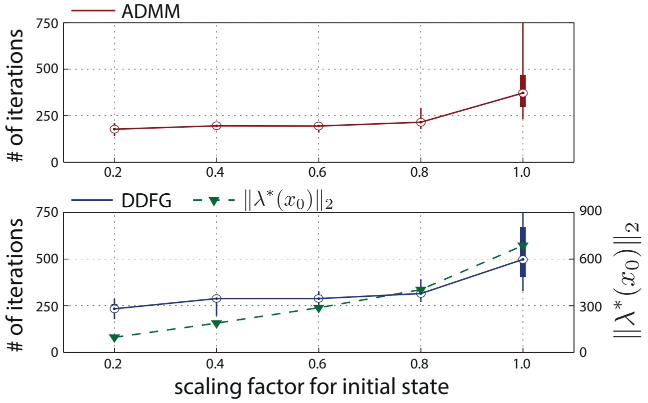


Fig. 5: Number of iterations to convergence and median $\|\lambda^*(x_0)\|_2$ under ADMM and DDFG for a network of 10 coupled double-integrators with 100 random initial states which were scaled by a factor in $\{0.2, 0.4, \ldots, 1.0\}$ (x-axis).

D. Number of Connected Systems

Scenario: We consider a chain of subsystems of form (17), with $\sigma_c^i = \sigma_s^i = 1 \ \forall i \in \mathcal{M}$. The number of connected elements takes values in $\{3, 50, 100, 150, 200, 250\}$ and for each size problem (6) is solved for 10 random initial conditions.

Findings: As depicted in Figure 6, the number of iterations to convergence increases with the size of the network. The increase is however significantly lower than linear and even seems to settle for a large number of elements. The value of $\|\lambda^*(x_0)\|_2$ correlates with the number of iterations to convergence.

Interpretation: The slope of $\|\lambda^*(x_0)\|_2$ closely resembles a square root function. This behavior can be explained by the fact that, as the length of the chain grows, the entries of the dual optimizer λ^* do not change in magnitude while only the dimension of λ^* becomes larger. Thus, $\|\lambda^*(x_0)\|_2$, grows with order $\mathcal{O}(\sqrt{M})$. From this observation and the existence of bound (14), we conjecture that the number of iterations, at least for DDFG, can be expected to grow according to order $\mathcal{O}(\sqrt{M})$ as well. Figure 6 suggests that the growth in the number of iterations may be even lower in practice.

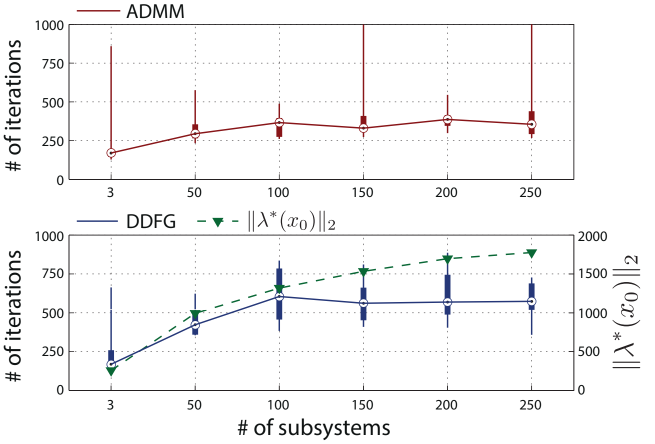


Fig. 6: Scaling behavior of distributed MPC for a chain of subsystems with $\sigma_c = \sigma_s = 1$ under ADMM and DDFG. The median of $\|\lambda^*(x_0)\|_2$ is depicted in the lower plot.

V. CONCLUSION

In this paper, the performance of two distributed optimization methods in MPC, namely DDFG and ADMM, was investigated by a systematic computational study. The analysis and results offer insight into the connection between fundamental attributes of distributed dynamic systems and the performance of distributed optimization methods.

The two attributes, to which the performance of the methods were the most sensitive, were coupling strength and stability of the subsystems. The methods however still perform well for coupling strengths which would lead to a very small region of attraction under non-iterative distributed MPC. Under distributed optimization in contrast, the region of attraction is the same as in a centralized approach. This underlines the much lower conservatism of iterative compared to non-iterative distributed MPC.

The performance of both methods was relatively insensitive to varying initial states. Even though initial states close to the boundary of the feasible set result in a slightly larger number of iterations, the methods are still tractable. This feature is beneficial for distributed MPC, where the initial state is the only problem parameter that changes during closed-loop operation. Thus, this finding suggests that distributed MPC will perform well during closed-loop operation.

Another conjecture is that iterative distributed MPC scales well with the size of the problem. More specifically for DDFG, as the number of subsystems increases, the increase in the number of iterations is expected to grow by no more than order $\mathcal{O}(\sqrt{M})$. This implies a great potential of iterative distributed MPC for large-scale systems where any centralized MPC approach eventually becomes intractable.

Regarding the methods that were compared, ADMM consistently showed a lower number of iterations to convergence than DDFG. Note furthermore, that there are no tuning parameters in DDFG, while the performance of ADMM might still be improved for a specific problem by the tuning parameter ρ.

REFERENCES

[1] W. B. Dunbar, "Distributed receding horizon control of dynamically coupled nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1249–1263, 2006.

[2] D. Jia and B. H. Krogh, "Distributed model predictive control," in *Proceedings of the American Control Conference*, Arlington, 2001, pp. 2767 – 2772.

[3] M. Farina and R. Scattolini, "Distributed non-cooperative mpc with neighbor-to-neighbor communication," in *Preprints of the 18th IFAC World Congress*, Milano, Italy, 2011, pp. 404–409.

[4] Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi, "Decentralized model predictive control via dual decomposition," in *Proceedings of the 47th Conference on Decision and Control*, Cancun, Mexico, 2008.

[5] I. Necoara, D. Doan, and J. A. K. Suykens, "Application of the proximal center decomposition method to distributed model predictive control," in *Proceedings of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

[6] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson, "On decentralized negotiation of optimal consensus," *Automatica*, vol. 44, pp. 1175–1179, 2008.

[7] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *44th IEEE Conference on Decisions and Control*, Seville, Spain, 2005, pp. 6680 – 6685.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[9] 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, 2000.

[10] C. Conte, N. R. Voellmy, M. N. Zeilinger, M. Morari, and C. N. Jones, "Distributed synthesis and control of constrained linear systems," in *Proceeding of the American Control Conference*, Montreal, Canada, 2012, accepted.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* Old Tappan, NJ (USA): Prentice Hall Inc., 1989.

[12] Y. Nesterov, *Introductory lectures on convex optimization.* Springer, 2004.

[13] S. Richter, M. Morari, and C. N. Jones, "Towards computational complexity certification for constrained mpc based on lagrange relaxation and the fast gradient method," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando,FL, 2011.