Automatica 69 (2016) 117–125

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Distributed synthesis and stability of cooperative distributed model predictive control for linear systems[☆]

Christian Conte [a,1], Colin N. Jones [b], Manfred Morari [a], Melanie N. Zeilinger [c]

[a] *Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zurich, 8092 Zurich, Switzerland*

[b] *Automatic Control Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland*

[c] *Institute for Dynamic Systems and Control, Department of Mechanical and Process Engineering, ETH Zurich, 8092 Zurich, Switzerland*

ARTICLE INFO

Article history:
Received 27 December 2013
Received in revised form
18 June 2015
Accepted 27 January 2016
Available online 19 March 2016

Keywords:
Distributed control
Predictive control
Large-scale systems

ABSTRACT

This paper presents a new formulation and synthesis approach for stabilizing cooperative distributed model predictive control (MPC) for networks of linear systems, which are coupled in their dynamics. The controller is defined by a network-wide constrained optimal control problem, which is solved online by distributed optimization. The main challenge is the definition of a global MPC problem, which both defines a stabilizing control law and is amenable to distributed optimization, i.e., can be split into a number of appropriately coupled subproblems. For such a combination of stability and structure, we propose the use of a separable terminal cost function, combined with novel time-varying local terminal sets. For synthesis, we introduce a method that allows for constructing these components in a completely distributed way, without central coordination. The paper covers the nominal case in detail and discusses the extension of the methodology to reference tracking. Closed-loop functionality of the controller is illustrated by a numerical example, which highlights the effectiveness of the proposed controller and its time-varying local terminal sets.

1. Introduction

Control of large-scale networks of dynamic systems is a challenging problem, in particular if the systems in the network are subject to communication constraints as well as constraints on states and inputs. MPC is a well-established methodology for the control of constrained systems. Its application under communication constraints has been a field of active research in recent years, with applications in fields such as power networks (Venkat, Hiskens, Rawlings, & Wright, 2008) and building automation (Ma, Richter, & Borrelli, 2012). Two key challenges in distributed MPC are closed-loop stability and controller synthesis under distributed computations. This paper addresses these challenges and proposes a less restrictive solution approach compared to methods currently available in the literature.

In order to obtain a distributed MPC formulation with stability guarantees, results from unconstrained *decentralized* and *distributed* control can be used. In decentralized control, the controllers in the network do not exchange information, while in distributed control they do. Important findings related to the analysis of decentralized systems are summarized in Šiljak (1991), where especially vector Lyapunov functions are used for stability analysis. Synthesis approaches for distributed control laws based on linear matrix inequalities have been proposed, e.g., in Langbort, Chandra, and D'Andrea (2004) and Zečević and Šiljak (2010).

The literature on distributed MPC mainly distinguishes between *non-cooperative* and *cooperative* approaches. In non-cooperative distributed MPC, e.g. Farina and Scattolini (2012), neighboring systems typically communicate once per time-step and each system is equipped with a local MPC controller that acts selfishly and is robust against coupling to neighboring systems. While requiring less communication, non-cooperative approaches can become very conservative or even infeasible in presence of strong dynamic coupling. In cooperative distributed MPC, as e.g. in

[☆] The work of C.N. Jones has received funding from the European Research Council under the European Unions Seventh Framework Programme (FP/2007–2013)/ ERC Grant Agreement no. 307608. The material in this paper was partially presented at the 2012 American Control Conference, June 27–29, 2012, Montreal, Canada and at the 52nd IEEE Conference on Decision and Control, December 10–13, 2013, Florence, Italy. This paper was recommended for publication in revised form by Associate Editor Faryar Jabbari under the direction of Editor Richard Middleton.

E-mail addresses: christian.conte@siemens.com (C. Conte), colin.jones@epfl.ch (C.N. Jones), morari@control.ee.ethz.ch (M. Morari), mzeilinger@ethz.ch (M.N. Zeilinger).

[1] Tel.: +41 78 6522388.

http://dx.doi.org/10.1016/j.automatica.2016.02.009
0005-1098/

Venkat, Rawlings, and Wright (2005) or more recently Giselsson and Rantzer (2013), neighboring systems typically communicate several times per time-step in order to solve a globally defined MPC problem by distributed optimization. A key requirement is for the MPC problem to be structured such that distributed optimization methods are applicable.

While some cooperative distributed MPC approaches derive stability guarantees based on long horizons (Giselsson & Rantzer, 2013), most approaches rely on terminal costs and terminal invariant sets (Mayne, Rawlings, Rao, & Scokaert, 2000). However, standard terminal costs and invariant sets are based on global Lyapunov stability and invariance concepts and do not exhibit a structure which is amenable to distributed optimization. In order to obtain such a structure for the terminal cost, vector Lyapunov functions (Šiljak, 1991) or linear matrix inequality (LMI) based methods (Langbort et al., 2004; Zečević & Šiljak, 2010) can be used. As for a structured terminal cost, however, the available methods are limited. One possibility is the use of a trivial terminal set, i.e. a point, as suggested in Stewart, Venkat, Rawlings, Wright, and Pannocchia (2010), which will, however, reduce the size of the region of attraction of the resulting MPC controller. Another option is to resort to robust positively invariant sets (Maestre, Muñoz de la Peña, Camacho & Alamo, 2011), considering dynamic coupling as a disturbance. In presence of strong coupling however, the resulting terminal sets tend to be small or may even be empty. Another possibility is the use of time-varying local sets, as suggested in Raković, Kern, and Findeisen (2010), the synthesis of which is however non-obvious and not discussed in the paper.

The contribution of this paper is twofold and builds on Conte, Voellmy, Zeilinger, Morari, and Jones (2012). The first contribution, based on Jokić and Lazar (2009), is a novel concept for time-varying local terminal sets leading to a cooperative distributed MPC controller with closed-loop stability guarantee. The proposed methodology can be used to construct terminal sets for both regulation and reference tracking MPC. As opposed to the concept presented in Raković et al. (2010), the set dynamics advocated in this paper are directly linked to the system dynamics and stability properties. The second contribution is a practical distributed synthesis method for networks of linear systems with quadratic costs and polytopic constraints. This method can be executed in a completely distributed way, a feature which is particularly beneficial in case of changing network topologies, where new controllers have to be synthesized on the fly without central coordination. This case has received considerable attention in the context of plug-and-play MPC (Riverso, Farina, & Trecate, 2013; Zeilinger, Pu, Riverso, Ferrari-Trecate, & Jones, 2013).

In Section 2, preliminaries on distributed systems and MPC are introduced. In Section 3, the formulation of the nominal cooperative distributed MPC controller is presented and in Section 4, its distributed synthesis is discussed. Section 5 summarizes the nominal case and Section 6 covers the extension to reference tracking. In Section 7, a numerical example is provided and Section 8 concludes the paper.

2. Preliminaries

2.1. Notation

The set $\{1,\dots,M\} \subseteq \mathbb{N}$ is denoted as $\mathcal{M}$. A block-diagonal matrix S with blocks S_i, where $i \in \mathcal{M}$, is denoted as $S = \mathrm{diag}_{i\in\mathcal{M}}(S_i)$ or $S = \mathrm{diag}(S_1,\dots,S_M)$, depending on the context. Similarly, a vector which consists of the stacked subvectors x_i, $i \in \mathcal{M}$, is denoted as $\mathrm{col}_{i\in\mathcal{M}}(x_i)$ or $\mathrm{col}(x_1,\dots,x_M)$. If a matrix S is positive definite, we write $S > 0$ and if it is positive semi-definite, we write $S \geq 0$. The n-dimensional identity matrix is denoted as I_n. A function $\beta(\cdot) : \mathbb{R}_+ \to \mathbb{R}_+$ is of class $\mathcal{K}$ if it is continuous, strictly increasing and if $\beta(0) = 0$. It is of class $\mathcal{K}_\infty$ if additionally it holds that $\lim_{s\to\infty} \beta(s) = \infty$.

2.2. Distributed linear time-invariant (LTI) Systems

We consider a network of M linear time-invariant systems, where each system $i \in \mathcal{M}$ has a state $x_i \in \mathbb{R}^{n_i}$, an input $u_i \in \mathbb{R}^{m_i}$ and an output $y_i \in \mathbb{R}^{p_i}$. We consider systems that are coupled in the state and in the output. The dynamics of the *local* systems can thus be written as

$$x_i^+ = \sum_{j=1}^{M} A_{ij}x_j + B_iu_i, \qquad y_i = \sum_{j=1}^{M} C_{ij}x_j \quad \forall i \in \mathcal{M}, \tag{1}$$

where $A_{ij} \in \mathbb{R}^{n_i\times n_j}$, $B_i \in \mathbb{R}^{n_i\times m_i}$ and $C_{ij} \in \mathbb{R}^{p_i\times n_j}$. Note that additional coupling in the inputs could be easily reformulated into the form in (1) by defining the original inputs as additional states and the changes in the original inputs as the new inputs. The coupling in states and outputs is used to define the notion of *neighboring systems*.

Definition 1 (*Neighboring Systems*)**.** System j is a neighbor of system i if $A_{ij} \neq 0$ or $C_{ij} \neq 0$. The set of all neighbors of i, including i itself, is denoted as $\mathcal{N}_i$. The states of all systems $j \in \mathcal{N}_i$ are denoted as $x_{\mathcal{N}_i} = \mathrm{col}_{j\in\mathcal{N}_i}(x_j) \in \mathbb{R}^{n_{\mathcal{N}_i}}$.

The local systems (1) can thus, with matrices of appropriate dimensions, be equivalently written as

$$x_i^+ = A_{\mathcal{N}_i}x_{\mathcal{N}_i} + B_iu_i, \qquad y_i = C_{\mathcal{N}_i}x_{\mathcal{N}_i} \quad \forall i \in \mathcal{M}. \tag{2}$$

Throughout the paper, it is assumed that neighboring systems can communicate with each other.

Assumption 2 (*Communication*)**.** Two systems i and j can communicate, in a bidirectional way, if $i \in \mathcal{N}_j$ or $j \in \mathcal{N}_i$.

Both the local states and inputs are subject to constraints

$$x_i \in \mathcal{X}_i, \qquad u_i \in \mathcal{U}_i \quad \forall i \in \mathcal{M}, \tag{3}$$

where for each $i \in \mathcal{M}$, $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ and $\mathcal{U}_i \subseteq \mathbb{R}^{m_i}$ are convex sets which contain the origin in their interior.

By combining the local system dynamics in (1), the linear dynamics of the *global* system result in

$$x^+ = Ax + Bu, \qquad y = Cx, \tag{4}$$

where $x = \mathrm{col}_{i\in\mathcal{M}}(x_i) \in \mathbb{R}^n$, $u = \mathrm{col}_{i\in\mathcal{M}}(u_i) \in \mathbb{R}^m$ and $y = \mathrm{col}_{i\in\mathcal{M}}(y_i) \in \mathbb{R}^p$. At some points in the paper, the equivalent notation $x^{t+1} = Ax^t + Bu^t$ will be used to emphasize the current time index. The global system matrix $A \in \mathbb{R}^{n\times n}$ and the global output map $C \in \mathbb{R}^{p\times n}$ are block-sparse with entries A_{ij} and C_{ij} for each $(i,j) \in \mathcal{M}^2$, provided $j \in \mathcal{N}_i$, and the global input map $B = \mathrm{diag}_{i\in\mathcal{M}}(B_i) \in \mathbb{R}^{n\times m}$ is block-diagonal. Similarly, combining the local state and input constraints (3), the global constraints for system (4) result in

$$\begin{aligned} x &\in \mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_M \subseteq \mathbb{R}^n, \\ u &\in \mathcal{U} := \mathcal{U}_1 \times \cdots \times \mathcal{U}_M \subseteq \mathbb{R}^m. \end{aligned} \tag{5}$$

In order for the methodology in this paper to apply, we make the following assumption on the stabilizability of (A, B).

Assumption 3 (*Structured Linear Controller*)**.** There exists a linear control law of the form

$$\kappa_f(x) := K_f x = \mathrm{col}_{i\in\mathcal{M}}(K_{\mathcal{N}_i}x_{\mathcal{N}_i}), \tag{6}$$

where $K_f \in \mathbb{R}^{m\times n}$ and $K_{\mathcal{N}_i} \in \mathbb{R}^{m_i\times n_{\mathcal{N}_i}}$ $\forall i \in \mathcal{M}$, such that the system $x^+ = Ax + B\kappa_f(x)$ is asymptotically stable.

Given a stabilizing controller $\kappa_f(x)$, the notion of a positively invariant set can be defined.