Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-23 (2015) 308–313

Cooperative Distributed Model Predictive Control via Linear Programming–A Divide and Conquer Approach ⋆

He Kong and Stefano Longo

Centre for Automotive Engineering and Technology, Cranfield University, College Road, Cranfield, Bedford MK43 0AL, United Kingdom. Email: h.kong@cranfield.ac.uk (He Kong), s.longo@cranfield.ac.uk (Stefano Longo)

Abstract: Motivated by the recent progress in centralized Model Predictive Control (MPC) design using a linear program (LP) and distributed MPC with quadratic program (QP) formulations, this paper discusses the design of LP-based cooperative distributed MPC for large scale linear systems with coupled dynamics and decoupled input constraints. As such, we examine the applicability of the Divide and Conquer approach to QP-based cooperative distributed MPC, recently proposed by the authors, to the LP-based case. It is shown that within the Divide and Conquer framework, an upbound of the original LP-based cooperative cost function (instead of itself), can be optimized for computing local inputs, resulting in suboptimal policies. However, advantages of doing so include the independency of the computation of local inputs and a large amount of communication burden reduction, as guaranteed by theoretical analysis and illustrated through numerical simulation. We also present closed-loop stability analysis and point out some questions worth further investigation.

Keywords: Predictive Control, Distributed Control, Constrained Control, Linear Programming

1. INTRODUCTION

MPC has received much consideration from both academia and industry (Mayne et al. (2000), Qin and Badgwell (2003)), due to its ability to handle multi-dimensional constrained systems with stability and robustness guarantee Goodwin et al. (2005), Borrelli et al. (2014), Rawlings and Mayne (2009), Goodwin et al. (2014), Lee (2014), and Mayne (2014). One tuning knob in MPC design is to formulate the cost either as a quadratic or linear criterion so that the optimization problem becomes a QP or LP, respectively. Compared to its QP formulation, MPC via LP is less studied, due to the elegant connection of the former to linear quadratic regulation theory. Nonetheless, design of optimal control and MPC via LP has received renewed attention recently (see, Rao and Rawlings (2000), Vandenberghe et al. (2002), Bemporad et al. (2002)), due to the development of more effective multiparametric LP methods Jones et al. (2007). Recent works have also shown the benefits of combining QP and LP in MPC to achieve sparsity Gallieri and Maciejowski (2012), Nagahara et al. (2014). Whilst these works provide great insights into LP-based MPC, the results are centralized in nature and may not be applicable to large scale systems with coupling dynamics, physical, and networked constraints.

In fact, there has been a recent rigorous interest to investigate distributed MPC paradigms Rawlings and Stewart (2008), Muller et al. (2012), Giselsson et al. (2013), Conte et al. (2013), Stewart et al. (2010). Such frameworks are desirable, since in centralized solutions, all subsystems rely on a central decision maker to coordinate and maintain plantwide actions, leading to organizational inefficiency, computational, and implementation issues. Existing distributed MPC methods can be classified as decentralized or distributed, cooperative or noncooperative, sequential or parallel, etc, Scattolini (2009), Christofides et al. (2013) and Maestre and Negenbaum (2014). An important discovery of the recent distributed MPC literature, among others, is that cooperative MPC emerges as the most attractive option among distributed MPC frameworks since it can guarantee closed-loop stability of a wide class of plantwide models *regardless* of the strength of the coupling dynamics, with a certain amount of communication burden increment Stewart et al. (2010). For an insightful discussion on different distributed MPC frameworks, one can refer to Chapter 6 of Rawlings and Mayne (2009).

We remark that most of the existing cooperative distributed MPC frameworks are QP-based and *suboptimal*, since the inherent characteristic of the cooperative question renders the computation of a single local input dependent on plantwide state information and the value of other local inputs. Very recently, in Kong et al. (2015), for the same problem setup with that of Stewart et al. (2010) (and Chapter 6 of Rawlings and Mayne (2009)), we have proposed a Divide and Conquer approach to cooperative distributed MPC. We have shown that the proposed approach with its QP formulation allows one to compute the local inputs independently with a moderate amount of communication burden. Besides, it works for any length of

⋆ This work was supported by the "Developing FUTURE Vehicles" project of the Engineering and Physical Sciences Research Council under the UK Low Carbon Vehicles Integrated Delivery Programme.

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2015.11.300

the prediction horizon and can be implemented in various ways to obtain the distributed local inputs, including explicit solutions, online parallel solutions without or with iteration, with closed-loop stability guaranteed *a priori*.

Motivated by the above progress, this paper discusses the design of LP-based cooperative distributed MPC for large scale linear systems with coupled dynamics and decoupled input constraints. Especially, we show that the Divide and Conquer approach to LP-based cooperative distributed MPC could enjoy those desirable features of its QP counterpart Kong et al. (2015). Although LP-based MPC might be solved more efficiently for large scale systems than its QP counterpart, we remark that our motivation to discuss the design of LP-based cooperative distributed MPC includes both architectural computation issues and communication considerations, among others. On the computation side, we believe that the development of LP-based distributed MPC frameworks can not only enrich the growing distributed MPC literature but also facilitate the applicability of existing LP-based centralized MPC results to large scale systems with coupling dynamics. On the communication side, for LP-based distributed MPC, it is generally true that certain information needs to be transmitted over uncertain networks, similarly with QP-based distributed MPC. However, there seems to be a sparse literature on LP-based control system design with networked induced effects, in contrast to the vast literature of QP-based design of optimal and predictive control of networked control systems Yang et al. (2014), Kong et al. (2014). This clearly leads to some less-studied questions worth consideration in the context of LP-based distributed MPC, e.g., robustness issues with networked effects. Thus, we remark that this paper shall not be considered as a minor generalization of Kong et al. (2015). It is also our hope that this work may raise more interest to investigate the design and real time implementation of LP-based distributed MPC Vichik and Borrelli (2014).

Notation: A^{T} stands for the transpose of matrix A and $\mathbf{R}^n$ stands for n-dimensional Euclidean space. $M > 0$ (≥ 0) means that M is real symmetric and positive definite (semi-definite). I stands for identity matrices of appropriate dimensions. Matrices with dimensions not being explicitly stated are assumed to be compatible for algebraic operations. $\mathscr{I}_{1:s}$ stands for the set of positive integers $\{1, \cdots, s\}$. $\mathrm{diag}\,[X, Y]$ denotes a block diagonal matrix with X, Y as its diagonal block entries.

2. PROBLEM FORMULATION

To illustrate, we adopt the system setup of Stewart et al. (2010). For simplicity, here we only consider the state feedback case with two subsystems.

2.1 About the models and the cost function with $1/\infty$ norm

We assume that for $(i, j) \in \mathscr{I}_{1:2} \times \mathscr{I}_{1:2}$, each subsystem i is a collection of the following linear discrete-time models

$$x_{ij}^+ = A_{ij} x_{ij} + B_{ij} u_j \tag{1}$$

where $A_{ij} \in \mathbf{R}^{n_{ij} \times n_{ij}}$, $B_{ij} \in \mathbf{R}^{n_{ij} \times m_j}$ are constant matrices; x_{ij}, u_j are the state vector and input vector, respectively, with u_j denoting the effects of input of subsystem j on the states of subsystem i. By collecting the states of subsystem 1 from (1), we obtain $x_1^+ = A_1 x_1 + \overline{B}_{11} u_1 + \overline{B}_{12} u_2$, with $x_1 = \left[\, x_{11}^{\mathrm{T}} \; x_{12}^{\mathrm{T}} \,\right]^{\mathrm{T}} \in \mathbf{R}^{n_1}$, $A_1 = \mathrm{diag}\,[A_{11}, A_{12}] \in \mathbf{R}^{n_1 \times n_1}$, $\overline{B}_{11} = \left[\, B_{11}^{\mathrm{T}} \; 0 \,\right]^{\mathrm{T}}, \overline{B}_{12} = \left[\, 0 \; B_{12}^{\mathrm{T}} \,\right]^{\mathrm{T}}$, where

$$\overline{B}_{1j} \in \mathbf{R}^{n_1 \times m_j}, \; n_1 = n_{11} + n_{12}. \tag{2}$$

The model of subsystem 2 can be obtained similarly. The plantwide model then becomes

$$x^+ = Ax + B_1 u_1 + B_2 u_2, \tag{3}$$

where $x = \left[\, x_1^{\mathrm{T}} \; x_2^{\mathrm{T}} \,\right]^{\mathrm{T}}, A = \mathrm{diag}\,[A_1, A_2], \; B_1 = \left[\, \overline{B}_{11}^{\mathrm{T}} \; \overline{B}_{21}^{\mathrm{T}} \,\right]^{\mathrm{T}}, \; B_2 = \left[\, \overline{B}_{12}^{\mathrm{T}} \; \overline{B}_{22}^{\mathrm{T}} \,\right]^{\mathrm{T}}$. For discussions on the relationship between the structures of the subsystems and the centralized model, one can refer to Stewart et al. (2010). It can be seen that both local inputs u_1 and u_2 have impact on the two subsystems. Denote $\mathbf{u}_1$ and $\mathbf{u}_2$ the local input sequences for the two subsystems along the prediction horizon, respectively. In LP-based cooperative distributed MPC, for each subsystem, a cost function based on a mixed $1/\infty$ norm, namely, 1-norm with respect to time and ∞-norm with respect to space is defined Bemporad et al. (2002). Each local input is required to minimize a global objective comprised of individual cost functions of the subsystems with relative weightings. To illustrate, for subsystem 1, we define the following cost function

$$V_1 = \sum_{k=0}^{N-1} [\|Q_1 x_1(k)\|_\infty + \|R_1 u_1(k)\|_\infty] + \|P_1 x_1(N)\|_\infty \tag{4}$$

where $Q_1 \in \mathbf{R}^{n_1 \times n_1}$ and $R_1 \in \mathbf{R}^{m_1 \times m_1}$ are nonsingular, $P_1 \in \mathbf{R}^{r_1 \times n_1}$ has full-column rank. For subsystem 2, we define a cost function V_2 in the same form of V_1. The plantwide objective function is defined to be

$$\mathscr{V} = \rho_1 V_1 + \rho_2 V_2 \tag{5}$$

where both ρ_1 and ρ_2 are positive real numbers representing the relative weights of V_1 and V_2, respectively. Note that the choice of ρ_1 and ρ_2 is problem dependent and can be considered as a design freedom for the user.

Remark 1. In cooperative distributed MPC, each local system computes its local input sequence by optimizing the plantwide objective function (5) (thereby accounting for the dynamics of the whole plant) and applies only the first move. Note that both V_1 and V_2 are implicit functions of both $\mathbf{u}_1$ and $\mathbf{u}_2$. Thus, the computation of the two local input sequences depends on each other. Also, as in centralized MPC, the choice of N and Q_i, R_i, P_i have influences on the solution to the question Kong et al. (2012), Kong et al. (2013).

2.2 About the constraints and assumptions

We assume that the local inputs are decoupled and satisfy $u_1(k) \in \mathbf{U}_1, u_2(k) \in \mathbf{U}_2$, for $k \in \mathscr{I}_{0:N-1}$, where $\mathbf{U}_1$ and $\mathbf{U}_2$ are compact and convex sets that include the origin in their interior. For the case of decoupled input constraints, the contribution of either local input on the cooperative cost function $\mathscr{V}$ cannot be affected by the other, and therefore can be neglected in both local computations. However, as remarked earlier, the computation of the local input sequence is dependent on each other. Thus, for $i \in \mathscr{I}_{1:2}$, the local optimization problems of computing $\mathbf{u}_i$ will be