Automatica 77 (2017) 31–43

Contents lists available at ScienceDirect

Automatica

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

Distributed predictive control with minimization of mutual disturbances[☆]

Paul A. Trodden[a,1], J.M. Maestre[b]

[a] Department of Automatic Control & Systems Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, UK

[b] Department of Systems and Automation Engineering, University of Seville, Seville, Spain

ARTICLE INFO

Article history:
Received 16 May 2016
Received in revised form
12 September 2016
Accepted 8 November 2016

Keywords:
Decentralization
Time-invariant
Control of constrained systems
Optimization-based controller synthesis
Parametric optimization

ABSTRACT

In this paper, a distributed model predictive control scheme is proposed for linear, time-invariant dynamically coupled systems. Uniquely, controllers optimize state and input constraint sets, and exchange information about these – rather than planned state and control trajectories – in order to coordinate actions and reduce the effects of the mutual disturbances induced via dynamic coupling. Mutual disturbance rejection is by means of the tube-based model predictive control approach, with tubes optimized and terminal sets reconfigured on-line in response to the changing disturbance sets. Feasibility and exponential stability are guaranteed under provided sufficient conditions on non-increase of the constraint set parameters.

1. Introduction

Model Predictive Control (MPC) has become one of the most popular advanced control techniques (Maciejowski, 2002), with many industrial applications (Qin & Badgwell, 2003) and mature theoretical foundations (Mayne, 2014). The key to this success is the inherent flexibility of MPC, which allows for complex issues such as constraints or delays to be dealt with explicitly, when otherwise the off-line determination of a control law would be prohibitively difficult. Despite this, the control of large-scale, interconnected or networked systems – such as chemical plants (Stewart, Venkat, Rawlings, Wright, & Pannocchia, 2010), electricity networks (McNamara, Negenborn, De Schutter, & Lightbody, 2013) or teams of vehicles (Trodden & Richards, 2013) – still presents significant difficulties to MPC (Negenborn & Maestre, 2014). For example, the organizational structure of the system – and its information flows – may not be conducive to a centralized control approach. Moreover, even if it is, the MPC optimization problem for the whole system may be too large to solve within the required time.

For this reason, significant attention has been given in the past decade to *distributed* forms of model predictive control (DMPC) (Christofides, Scattolini, Muñoz del la Peña, & Liu, 2013; Maestre & Negenborn, 2014; Scattolini, 2009). In DMPC, the optimal control problem is decomposed into several smaller sub-problems that are distributed to a set of local controllers or *control agents*. Each controller or agent is responsible for controlling a subsystem composed of a subset of the system states and control inputs. In order to achieve system-wide stability and satisfactory closed-loop performance, the agents exchange information so that they can coordinate their decision making. Many schemes have been proposed to date, and differ according to the particularities of the scenarios in which they are applied: for example, the way in which the system is decomposed, the source of coupling, or the limits in the communication or computation capacity (Maestre & Negenborn, 2014).

One of the fundamental, and most researched, problems in DMPC is control of linear time-invariant systems coupled via dynamics. The problem is non-trivial since the states and inputs of one subsystem affect others too, leading to mutual disturbances; hence, coordination is usually needed to ensure satisfactory performance of the overall system. Many approaches have been proposed (Christofides et al., 2013; Maestre & Negenborn, 2014; Scattolini, 2009), and almost all involve the sharing of planned

[☆] J. M. Maestre is supported by the Spanish Ministry of Economy and Competitiveness (project COOPERA under grant DPI2013-46912-C2-1-R). The material in this paper was partially presented at the 2016 American Control Conference, July 6–8, 2016, Boston, MA, USA. This paper was recommended for publication in revised form by Associate Editor Riccardo Scattolini under the direction of Editor Ian R. Petersen.

E-mail addresses: p.trodden@sheffield.ac.uk (P.A. Trodden), pepemaestre@us.es (J.M. Maestre).

[1] Fax: +44 0 114 2225683.

http://dx.doi.org/10.1016/j.automatica.2016.11.023
0005-1098/

control sequences or state trajectories between controllers. Recently, attention has focused on tube MPC (Mayne, Seron, & Raković, 2005) as a means for rejecting the mutual disturbances arising from these subsystem interactions. The first tube-based DMPC approaches (Richards & How, 2007; Trodden & Richards, 2010) were developed for dynamically *decoupled*, uncertain subsystems with coupled constraints; each controller uses the tube technique to reject bounded local disturbances. The direct application of that approach to systems with dynamic coupling will, however, result in excessive conservativeness, since the bounded disturbance set for each subsystem must account for *all* possible state and input interactions (and not just, for example, deviations of neighbours' states and inputs from planned, or reference, trajectories). To circumvent this, improved proposals have been made: in Farina and Scattolini (2012), tube-based controllers share reference trajectories and maintain true states and inputs in bounded neighbourhoods of these. In Riverso and Ferrari-Trecate (2012), the tube MPC concept is applied *twice* by each controller: once to maintain a planned perturbed state trajectory around a planned nominal trajectory, then again to maintain the true, perturbed state trajectory around the planned one.

Though providing a natural route to guaranteed feasibility and stability, a key drawback of the tube-based approaches is conservatism because, ultimately, the mutual disturbance induced by state and/or input coupling has to be bounded. If the state and input constraint sets are large, then this naturally leads to large disturbance sets and, hence, more tightly constrained local optimal control problems, even for (Farina & Scattolini, 2012; Riverso & Ferrari-Trecate, 2012). In this paper, we attempt to overcome this drawback by exploiting the fact that, often, subsystems do not *use* all of their state and input constraint sets and, hence, the mutual disturbance sets can be reduced by considering this. The main technical development is that local controllers, when solving their optimal control problems, optimize not only the control sequence but also the *sizes* of the state and input constraint sets. In other words, subsystem state and input sets are contracted to the smallest sizes sufficient to meet control objectives, which in turn leads to smaller disturbance sets. Controllers then share information about these state and input sets – rather than planned state and control trajectories – in order that they may compute a smaller estimate of the set of possible disturbances. Finally, to reject these bounded disturbances, the tube MPC technique (Mayne et al., 2005) is applied. However, in this paper, the disturbance invariant sets required for tube MPC are optimized online to take into account the changing sizes of the disturbance sets.

The sharing of *sets* of states and inputs has similarities with the "contract-based" DMPC approach (Lucia, Kögel, & Findeisen, 2015), wherein subsystems share "contract sets" about their future behaviour, based on reachable sets computed at each time step given current knowledge of uncertainty. Our work differs in several details, including (i) the use of decoupled positively invariant sets as terminal conditions, which are less complex objects, and easier to compute, than the inter-dependent robust invariant sets required in Lucia et al. (2015); (ii) in our approach, the complexity of each MPC problem is similar to conventional MPC, and the shared information between subsystems is of parameterized versions of the state and input constraint sets, which are readily available, while in Lucia et al. (2015) sequences of reachable sets are required to be computed within each MPC optimization; (iii) we offer a comprehensive way to compute the required disturbance sets and robust invariant sets that arise from the shared state and input sets, via a single linear program (LP).

This latter aspect, in particular, of the proposed approach also leads to similarities with the "plug-and-play" approach to decentralized MPC (Riverso, Farina, & Ferrari-Trecate, 2014). In that approach, subsystem controllers re-compute disturbance invariant sets on-line in order to account for changes to disturbance sets. However, there are two key differences: firstly, in Riverso et al. (2014), only the effect of adding or removing subsystems from the overall system is considered when disturbance sets are re-computed, while in this paper we re-compute disturbance sets to account for how much of the constraint sets planned state and input trajectories are using. Secondly, in Riverso et al. (2014) the notion of robust control invariant (RCI) sets (Raković, Kerrigan, Mayne, & Kouramas, 2007) is used: each subsystem controller solves an LP to compute an RCI set and an associated feedback control law which are then used as, respectively, the tube cross-section set and tube controller. In this paper, however, we retain the original notion in tube MPC of robust positively invariant (RPI) sets: each controller retains the same (linear) tube controller throughout, but solves an LP to re-compute its RPI tube cross-section set to take into account changes to the mutual disturbance set. This is achieved by exploiting a recently developed method for computing, via a single LP, an RPI set characterized by *a-priori* known inequalities (Trodden, in press); we make a further extension to this approach to include the computation of the disturbance set (which depends on neighbouring subsystems' states and inputs) implicitly in the RPI set optimization, removing the need to compute the disturbance set explicitly beforehand.

A preliminary version of this paper appeared in Trodden, Baldivieso, and Maestre (2016), presenting the initial idea and results. In the current paper, the following additional contributions are made:

- A reconfigurable, parametric terminal set is designed, replacing the simple choice of the origin used in Trodden et al. (2016). This set, which enlarges the region of attraction and improves closed-loop performance, adjusts automatically (on-line) to account for the changes in size and shape of the constraint sets.
- The ancillary on-line operations to re-compute disturbance invariant sets are refined and improved: RPI sets are computed directly from shared information, via a single LP, removing the need to explicitly construct disturbance sets via Minkowski summations as in Trodden et al. (2016). Furthermore, the algorithm is generalized to permit re-configuration of sets at a lower rate than the main sampling rate, in order to reduce the on-line computational burden. Further simplifications are described and discussed, including a scalar implementation of the algorithm that requires minimal on-line computation in addition to the MPC problem.

The paper is organized as follows. Preliminary details and the problem statement are given in Section 2. In Section 3, the distributed optimal control problem, including the parametric design of the terminal set, is presented. The distributed control algorithm is defined in Section 4, together with details and explanations of on-line computations. Theoretical guarantees of recursive feasibility and stability, under the sufficient condition of non-increase of the state and input constraint set parameters, are established in Section 5. In Section 6, simulations of the algorithm are presented for an example system, before concluding remarks are made in Section 7.

Notation: The sets of non-negative and positive reals are denoted, respectively, $\mathbb{R}_{0+}$ and $\mathbb{R}_{+}$. The notation $[a, b]^n$ means the n-dimensional product set $[a, b] \times [a, b] \times \cdots \times [a, b]$, where $a \in \mathbb{R}$ and $b \in \mathbb{R}$. For $a, b \in \mathbb{R}^n$, $a \leq b$ applies element by element. The ball of radius δ is $\mathcal{B}(\delta)$; the dimension will be clear from the context. The distance of a point $x \in \mathbb{R}^n$ from a set $X \subset \mathbb{R}^n$ is $|x|_X \triangleq \inf_{y\in X} |x - y|$. AX denotes the image of a set $X \subset \mathbb{R}^n$ under the linear mapping $A : \mathbb{R}^n \mapsto \mathbb{R}^p$, and is given by $\{Ax : x \in X\}$. For $X, Y \subset \mathbb{R}^n$, the Minkowski sum is $X \oplus Y \triangleq \{x+y : x \in X, y \in Y\}$; for $Y \subset X$, the Minkowski difference is $X \ominus Y \triangleq \{x \in \mathbb{R}^n : Y \oplus \{x\} \subset X\}$. For $X \subset \mathbb{R}^n$ and $a \in \mathbb{R}^n$, $X \oplus a$ means $X \oplus \{a\}$. The support function