Article:

Trodden, P. and Richards, A. (2010) Distributed model predictive control of linear systems with persistent disturbances. International Journal of Control, 83 (8). 1653 - 1663. ISSN 0020-7179

https://doi.org/10.1080/00207179.2010.485280

International Journal of Control
Vol. 00, No. 00, Month 200x, 1–15

RESEARCH ARTICLE

Distributed model predictive control of linear systems with persistent disturbances

Paul Trodden* & Arthur Richards

Department of Aerospace Engineering, University of Bristol, Bristol BS8 1TR, United Kingdom

(*v2 released April 2010*)

This paper presents a new form of robust distributed model predictive control (MPC) for multiple dynamically-decoupled subsystems, in which distributed control agents exchange plans to achieve satisfaction of coupling constraints. The new method offers greater flexibility in communications than existing robust methods, and relaxes restrictions on the order in which distributed computations are performed. The local controllers use the concept of tube MPC – in which an optimization designs a tube for the system to follow rather than a trajectory – to achieve robust feasibility and stability despite the presence of persistent, bounded disturbances. A methodical exploration of the trades between performance and communication is provided by numerical simulations of an example scenario. It is shown that at low levels of inter-agent communication, the DMPC can obtain a lower closed-loop cost than that obtained by a centralized implementation. A further example shows that the flexibility in communications means the new algorithm has a relatively low susceptibility to the adverse effects of delays in computation and communication.

Keywords: linear systems, distributed control, constrained control

1 Introduction

This paper develops a distributed form of Model Predictive Control (MPC) (Mayne et al. 2000, Maciejowski 2002) for a group of linear subsystems that guarantees stability and satisfaction of coupled constraints despite the action of persistent, unknown, but bounded disturbances. The distributed control agents communicate plans with each other to achieve constraint satisfaction. Key features of the new formulation are that (i) only one subsystem agent updates its plan at each time step, (ii) robust stability is guaranteed for any choice of update sequence, and (iii) each agent communicates only after its update; the resulting algorithm offers flexibility in communication and computation. This is the first work to combine guaranteed robust feasibility and convergence, in the presence of a persistent disturbance, with flexible communication. In addition, this paper presents a thorough investigation of the trade between performance and communication for an example scenario, identifying how to exploit the flexibility of the new algorithm, and examines the effects on performance of delays in communication and computation.

Decentralized or *Distributed* MPC (DMPC) (Camponogara et al. 2002) has been developed for application to large-scale systems, such as chemical plants (Venkat et al. 2004) and process control (Borrelli et al. 2005), or teams of vehicles (Kuwata et al. 2007), in which a control by a single centralized agent would require excessive communication, computation and reliance on a single processor. Instead, DMPC distributes control decision-making among agents corresponding to the different subsystems making up the whole. The challenge is then how to coordinate efforts to ensure that the distributed decisions lead to constraint satisfaction, feasibility and stability of the overall closed-loop system.

*Corresponding author. Email: paul.trodden@bris.ac.uk

ISSN: 0020-7179 print/ISSN 1366-5820 online

DOI: 10.1080/0020717YYxxxxxxxx
http://www.informaworld.com

Several strategies for DMPC have been presented in the literature, and many theoretical results exist, including those for feasibility and stability; see Scattolini (2009) for a comprehensive survey. The approaches are broadly divisible by the type of couplings or interactions assumed between constituent subsystems. For example, dynamically-coupled systems (Du et al. 2001, Camponogara et al. 2002, Ling et al. 2005, Dunbar 2007, Giovanini et al. 2007, Venkat et al. 2008), coupling via the cost function (Shim et al. 2003, Raffard et al. 2004, Franco et al. 2007), and subsystems sharing coupled constraints (Waslander et al. 2004, Keviczky et al. 2006, Richards and How 2007, Kuwata et al. 2007). The method presented in this paper assumes the latter type of coupling, and has agents update their plans one at a time, without iteration, to ensure coupled constraint satisfaction; however, unlike other methods, it also permits a flexible order of updating.

Robustness to disturbances is a key challenge in the development of MPC (Mayne et al. 2000), and is harder still when control decision-making is decentralized; few DMPC schemes in the literature offer robustness . In Richards and How (2007), robust feasibility and stability are guaranteed by updating each subsystem's plan in a sequence, subject to tightened constraints, and while 'freezing' the plans of others. Alternative approaches include treatment of interconnected subsystems' state trajectories as bounded uncertainties, and using min-max optimization (Jia and Krogh 2002) – though the complexity issues with such an optimization method are well documented (Mayne et al. 2000). Using the comparison model approach to robustness (Fukushima and Bitmead 2005), another distributed method (Kim and Sugie 2005) uses worst-case predictions of state errors, determined based on a robust control Lyapunov function, and tightens constraints accordingly. Magni and Scattolini (2006) propose a robust stable decentralized algorithm for non-linear dynamically-coupled systems, with no information exchange between agents, although for an asymptotically-decaying disturbance.

The distributed MPC method presented in this paper achieves robustness to persistent disturbances by use of *tube MPC* (Mayne et al. 2005), a form of robust MPC that guarantees feasibility and stability despite the action of an unknown but bounded disturbance. In this formulation, the 'tube' is a sequence of robust invariant sets centered on a trajectory for the nominal (*i.e.*, disturbance-free) system; use of feedback ensures that the system remain insides the tube for all possible realizations of the disturbance. A key observation of this new work is that if that feedback uses only local information, each subsystem can remain within its tube without the need for communication, and exchange of information with other agents is only required when the tubes are updated by the optimization. The new algorithm in this paper exploits this feature to achieve flexibility in communication. An additional advantage of this approach is that the optimization involves only the nominal system dynamics, avoiding the large increase in computational complexity associated with the inclusion of uncertainty in the optimization (Scokaert and Mayne 1998).

Many distributed methods proposed in the literature (*e.g.*, Du et al. (2001), Kim and Sugie (2005), Dunbar and Murray (2006), Alessio and Bemporad (2007), Richards and How (2007), Venkat et al. (2008)) do not consider the implications that the scheduling of local optimizations has on the time required for communications. For example, the *constraint-tightening* DMPC approach proposed by Richards and How (2007), also for dynamically-decoupled systems with coupled constraints, assumes repeated *instantaneous* exchanges during each sampling period. On the other hand, Jia and Krogh (2002) used a stability constraint to permit a one-step delay in information exchange, while Franco et al. (2007, 2008) show input-to-state stability for systems with for multiple-step delays. Richards and How (2005) present a robust DMPC method with explicit allowance for computation and communication delays. Though delays are not explicitly considered for the new algorithm developed in this paper, its single-update nature implicitly allows time for communications after each optimization, and *no* instantaneous inter-agent exchanges of information are assumed. We provide a numerical investigation of the effects of delays on the new algorithm; the results highlight the reduced susceptibility of the proposed tube DMPC to delay in both communication and computation.

Section 2 defines the problem statement, and reviews tube MPC. Section 3 develops the main result, a robust distributed MPC algorithm, by extending tube MPC to a distributed implementation where only one subsystem agent updates at each time step. Section 4 analyses the communication requirements for the new algorithm, and Section 5 presents results from numerical simulations, including an exploration of the trades between performance and communication, and an investigation into the effects of delays.

Notation: The matrix mapping of a set is defined as $\mathbf{A}\mathcal{B} \triangleq \{\mathbf{c} \mid \exists \mathbf{b} \in \mathcal{B}, \mathbf{c} = \mathbf{A}\mathbf{b}\}$. The operator '$\sim$' denotes the Pontryagin difference (Kolmanovsky and Gilbert 1998), a set-shrinking operation defined as $\mathcal{A} \sim \mathcal{B} \triangleq \{\mathbf{a} \mid \mathbf{a}+\mathbf{b} \in \mathcal{A}, \forall \mathbf{b} \in \mathcal{B}\}$. The operator '$\oplus$' denotes the Minkowski sum, defined as $\mathcal{A} \oplus \mathcal{B} \triangleq \{\mathbf{a}+\mathbf{b}, \mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}\}$. The double subscript notation $(k+j|k)$ indicates a prediction of a variable j steps ahead from time k. Let $\mathbb{N} \triangleq \{0, 1, 2, \ldots\}$.

2 Preliminaries

2.1 *Problem statement*

The aim is to control a system of N_p linear time-invariant, discrete-time subsystems, the set of which is denoted $\mathcal{P} = \{1, \ldots, N_p\}$, described by the state equations

$$\mathbf{x}_p(k+1) = \mathbf{A}_p\mathbf{x}_p(k) + \mathbf{B}_p\mathbf{u}_p(k) + \mathbf{w}_p(k), \forall p \in \mathcal{P}, k \in \mathbb{N}, \tag{1}$$

where $\mathbf{x}_p \in \mathbb{R}^{N_{x,p}}$, $\mathbf{u}_p \in \mathbb{R}^{N_{u,p}}$ and $\mathbf{w}_p \in \mathbb{R}^{N_{x,p}}$ are respectively the state vector, control input vector, and disturbance acting on subsystem p. Assume that each system $(\mathbf{A}_p, \mathbf{B}_p)$ is controllable, and that the complete states $\mathbf{x}_p$ are available at each sampling instant. The disturbances are unknown *a priori*, but are assumed to lie in known independent compact sets that contain the origin:

$$\mathbf{w}_p(k) \in \mathcal{W}_p \subset \mathbb{R}^{N_{x,p}}, \forall p \in \mathcal{P}, k \in \mathbb{N}.$$

Each subsystem is subject to local constraints:

$$\mathbf{C}_p\mathbf{x}_p(k) + \mathbf{D}_p\mathbf{u}_p(k) \in \mathcal{Y}_p \subset \mathbb{R}^{N_{y,p}}, \forall p \in \mathcal{P}, k \in \mathbb{N},$$

where $\mathcal{Y}_p$ is closed, and also N_c coupling constraints across multiple subsystems. Each coupling constraint $c \in \mathcal{C} = \{1, \ldots, N_c\}$ applies to the sum of coupling outputs $\mathbf{z}_{cp} \in \mathbb{R}^{N_{z,c}}$:

$$\forall c \in \mathcal{C}, p \in \mathcal{P}, k \in \mathbb{N} : \mathbf{z}_{cp}(k) = \mathbf{E}_{cp}\mathbf{x}_p(k) + \mathbf{F}_{cp}\mathbf{u}_p(k),$$
$$\sum_{p=1}^{N_p} \mathbf{z}_{cp}(k) \in \mathcal{Z}_c \subset \mathbb{R}^{N_{z,c}},$$

where $\mathcal{Z}_c$ is closed. The matrices $\mathbf{C}_p, \mathbf{D}_p, \mathbf{E}_{cp}, \mathbf{F}_{cp}$ and the sets $\mathcal{Y}_p, \mathcal{Z}_c$ are all chosen by the designer as part of the problem.

The system-wide objective is assumed to be decoupled, and is a summation of some function of the state and input, given by

$$\min \sum_{p=1}^{N_p} \sum_{k=0}^{\infty} l_p\big(\mathbf{x}_p(k), \mathbf{u}_p(k)\big), \tag{2}$$

where it is assumed that $l_p\big(\mathbf{x}_p, \mathbf{u}_p\big) \geq c \left\|\mathbf{x}_p, \mathbf{u}_p\right\|$ for some $c > 0$, and $l_p\big(\mathbf{0}, \mathbf{0}\big) = \mathbf{0}$.

2.2 *Coupling structure*

The following definitions identify structure in the coupling, and are used later to determine the requirements for communication. Define $\mathcal{P}_c$ as the set of all subsystems involved in constraint c, and similarly let $\mathcal{C}_p$ be the set of constraints involving subsystem p:

$$\mathcal{P}_c \triangleq \Big\{p \in \mathcal{P} : \big[\mathbf{E}_{cp}\,\mathbf{F}_{cp}\big] \neq \mathbf{0}\Big\}, \tag{3}$$

$$\mathcal{C}_p \triangleq \Big\{c \in \mathcal{C} : \big[\mathbf{E}_{cp}\,\mathbf{F}_{cp}\big] \neq \mathbf{0}\Big\}. \tag{4}$$

Then the set of all other subsystems coupled to p is

$$\mathcal{Q}_p = \left(\bigcup_{c \in \mathcal{C}_p} \mathcal{P}_c\right) \backslash \{p\}. \tag{5}$$

2.3 *Tube model predictive control*

Tube MPC (Mayne et al. 2005) uses the *nominal* system dynamics to design a sequence of disturbance-invariant state sets for a horizon of N steps. The decision variable includes the initial state, and is defined as $\mathbf{U}_p(k) \triangleq \big\{\bar{\mathbf{x}}_p(k|k), \bar{\mathbf{u}}_p(k|k), \ldots, \bar{\mathbf{u}}_p(k+N-1|k)\big\}, \forall p \in \mathcal{P}$. As the optimization involves only nominal terms, complexity is comparable to standard MPC, and robustness to disturbance is guaranteed by use of a feedback law to keep the state around the tube centre. The following standing assumption is required: there exists a local stabilizing controller $\mathbf{K}_p$ for each subsystem $\big(\mathbf{A}_p, \mathbf{B}_p\big)$ and hence a corresponding robust positively-invariant (RPI) set $\mathcal{R}_p$, satisfying

$$\begin{aligned} \big(\mathbf{A}_p + \mathbf{B}_p\mathbf{K}_p\big)\mathbf{x}_p + \mathbf{w}_p \in \mathcal{R}_p, \forall \mathbf{x}_p \in \mathcal{R}_p, \mathbf{w}_p \in \mathcal{W}_p, \\ \big(\mathbf{C}_p + \mathbf{D}_p\mathbf{K}_p\big)\mathcal{R}_p \subset \mathcal{Y}_p, \\ \bigoplus_{p=1}^{N_p}\big(\mathbf{E}_{cp} + \mathbf{F}_{cp}\mathbf{K}_p\big)\mathcal{R}_p \subset \mathcal{Z}_c, \forall c \in \mathcal{C}. \end{aligned} \tag{6}$$

Then the *centralized* problem $\mathbb{P}^{\mathrm{C}}\big(\mathbf{x}_1(k), \ldots, \mathbf{x}_{N_p}(k)\big)$ is

$$J^{\mathrm{opt}}\big(\mathbf{x}_1(k), \ldots, \mathbf{x}_{N_p}(k)\big) = \min_{\{\mathbf{U}_1(k), \ldots, \mathbf{U}_{N_p}(k)\}} \sum_{p=1}^{N_p} J_p\big(\mathbf{U}_p(k)\big) \tag{7}$$

subject to $\forall p \in \mathcal{P}, \forall j \in \{0, \ldots, N-1\}$:

$$\bar{\mathbf{x}}_p(k+j+1|k) = \mathbf{A}_p\bar{\mathbf{x}}_p(k+j|k) + \mathbf{B}_p\bar{\mathbf{u}}_p(k+j|k), \tag{8a}$$

$$\mathbf{x}_p(k) - \bar{\mathbf{x}}_p(k|k) \in \mathcal{R}_p, \tag{8b}$$

$$\bar{\mathbf{x}}_p(k+N|k) \in \mathcal{X}_{F_p}, \tag{8c}$$

$$\bar{\mathbf{y}}_p(k+j|k) = \mathbf{C}_p\bar{\mathbf{x}}_p(k+j|k) + \mathbf{D}_p\bar{\mathbf{u}}_p(k+j|k), \tag{8d}$$

$$\bar{\mathbf{y}}_p(k+j|k) \in \tilde{\mathcal{Y}}_p, \tag{8e}$$

$$\forall c \in \mathcal{C}: \quad \bar{\mathbf{z}}_{cp}(k+j|k) = \mathbf{E}_{cp}\bar{\mathbf{x}}_p(k+j|k) + \mathbf{F}_{cp}\bar{\mathbf{u}}_p(k+j|k), \tag{8f}$$

$$\sum_{p=1}^{N_p} \bar{\mathbf{z}}_{cp}(k+j|k) \in \tilde{\mathcal{Z}}_c, \tag{8g}$$

where the cost function is a finite-horizon approximation to (2), involving the nominal states and inputs:

$$J_p\big(\mathbf{U}_p(k)\big) \triangleq F_p\big(\bar{\mathbf{x}}_p(k+N|k)\big) + \sum_{j=0}^{N-1} l_p\big(\bar{\mathbf{x}}_p(k+j|k), \bar{\mathbf{u}}_p(k+j|k)\big). \tag{9}$$

The sets $\tilde{\mathcal{Y}}_p, \tilde{\mathcal{Z}}_c$ represent the sets $\mathcal{Y}_p, \mathcal{Z}_c$ tightened by margins to allow for uncertainty:

$$\tilde{\mathcal{Y}}_p = \mathcal{Y}_p \sim \big(\mathbf{C}_p + \mathbf{D}_p\mathbf{K}_p\big)\mathcal{R}_p, \tag{10a}$$

$$\tilde{\mathcal{Z}}_c = \mathcal{Z}_c \sim \bigoplus_{p=1}^{N_p} \big(\mathbf{E}_{cp} + \mathbf{F}_{cp}\mathbf{K}_p\big)\mathcal{R}_p. \tag{10b}$$

The sets $\mathcal{R}_p$ are 'cross-sections' of the tubes and are RPI sets, as in (6). The sets $\mathcal{X}_{F_p}$ are terminal sets, each assumed to have an interior, and invariant under terminal control laws $\mathbf{u}_p = \kappa_{F_p}(\mathbf{x}_p)$, $\forall p \in \mathcal{P}$, so that for all $\mathbf{x}_p \in \mathcal{X}_{F_p}$,

$$\mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\kappa_{F_p}(\mathbf{x}_p) \in \mathcal{X}_{F_p}, \tag{11a}$$

$$\mathbf{C}_p\mathbf{x}_p + \mathbf{D}_p\kappa_{F_p}(\mathbf{x}_p) \in \tilde{\mathcal{Y}}_p, \tag{11b}$$

$$\sum_{p=1}^{N_p} \mathbf{E}_{cp}\mathbf{x}_p + \mathbf{F}_{cp}\kappa_{F_p}(\mathbf{x}_p) \in \tilde{\mathcal{Z}}_c, \forall c \in \mathcal{C}. \tag{11c}$$

A further assumption is that, for each p, the terminal cost is a local Lyapunov function in $\mathcal{X}_{F_p}$:

$$F_p\big(\mathbf{A}_p\mathbf{x}_p + \mathbf{B}_p\kappa_{F_p}(\mathbf{x}_p)\big) - F_p\big(\mathbf{x}_p\big) \le -l_p\big(\mathbf{x}_p, \kappa_{F_p}(\mathbf{x}_p)\big), \forall \mathbf{x}_p \in \mathcal{X}_{F_p}, p \in \mathcal{P}. \tag{12}$$

Assumptions (11) and (12), together with the requirements on the stage cost, represent A1–A4 in Mayne et al. (2000) or equivalently A1 and A2 in Mayne et al. (2005).

After the optimization is solved at each time step, the following control is applied to each subsystem $p \in \mathcal{P}$

$$\mathbf{u}_p(k) = \bar{\mathbf{u}}_p^{\text{opt}}(k|k) + \mathbf{K}_p\big(\mathbf{x}_p(k) - \bar{\mathbf{x}}_p^{\text{opt}}(k|k)\big). \tag{13}$$

Under this control, the closed-loop system then is robustly-feasible and stable; see Mayne et al. (2005, Proposition 3).

3 Robust distributed MPC using tubes

This section extends tube MPC (Mayne et al. 2005) to a distributed implementation, with application to the problem statement in Section 2, and states the main feasibility and stability results. The centralized problem $\mathbb{P}^{\mathrm{C}}$ is distributed amongst subsystem agents as local optimization problems, and only one subsystem is permitted to update at each time step; it is possible to permit the simultaneous updating of all agents in some cases (Trodden 2009), although this generalization is not considered here. In the sequel, p_k shall denote the agent optimizing at time k. Therefore, how an agent obtains a new plan depends on whether it is selected for update: if $p = p_k$, the new plan for p is obtained as the solution to the local optimization; otherwise, the previous plan for p is renewed by taking the tail of the previous feasible solution and augmenting with a step of terminal control κ_{F_p}. That is, given $\mathbf{U}_p^*(k)$ at time k,

$$\tilde{\mathbf{U}}_p(k+1) \triangleq \big\{\bar{\mathbf{x}}_p^*(k+1|k), \bar{\mathbf{u}}_p^*(k+1|k), \ldots, \bar{\mathbf{u}}_p^*(k+N-1|k), \kappa_{F_p}\big(\bar{\mathbf{x}}_p^*(k+N|k)\big)\big\}, \tag{14}$$

is a feasible plan for time $k+1$. The agents thus update in a sequence, $\big\{p_1, \ldots, p_k, p_{k+1}, \ldots\big\}$, to be chosen by the designer. The local problem $\mathbb{P}_p^{\mathrm{D}}\big(\mathbf{x}_p(k); \mathbf{Z}_p^*(k)\big)$ for a subsystem $p \in \mathcal{P}$ is defined by

$$J_p^{\mathrm{opt}}\big(\mathbf{x}_p(k); \mathbf{Z}_p^*(k)\big) = \min_{\mathbf{U}_p(k)} J_p\big(\mathbf{U}_p(k)\big) \tag{15}$$

subject to constraints (8a) to (8f) for agent p only, and

$$\bar{\mathbf{z}}_{cp}(k+j|k) + \sum_{q \in \mathcal{P}_c \setminus \{p\}} \bar{\mathbf{z}}_{cq}^*(k+j|k) \in \tilde{\mathcal{Z}}_c, \forall c \in \mathcal{C}_p. \tag{16}$$

In this optimization, $\mathbf{Z}_p^*(k)$ denotes the collection of outputs $\bar{\mathbf{z}}_{cq}^*(\cdot|k)$ required by p to evaluate constraint (16). Note that the collection of (16) over all subsystems $p \in \mathcal{P}$ is equivalent to (8g); the revised summation removes terms that are identically zero, using the definitions (3) and (4). We assume at this point that the information $\mathbf{Z}_p^*(k)$ is known; in Section 4 the communication requirements to obtain $\mathbf{Z}_p^*(k)$ are identified. This local optimization is then employed in the following algorithm, executed by all agents in parallel.

Algorithm 1:

(i) Set $k = 0$. Wait for feasible solution $\mathbf{U}_p^*(0)$, information $\mathbf{Z}_p^*(0)$, and terminal set $\mathcal{X}_{F_p}$ and control law κ_{F_p} from central initializing agent.
(ii) Apply control (13): $\mathbf{u}_p(k) = \bar{\mathbf{u}}_p(k|k) + \mathbf{K}_p\big(\mathbf{x}_p(k) - \bar{\mathbf{x}}_p(k|k)\big)$.
(iii) Increment k, and sample current state $\mathbf{x}_p(k)$.
(iv) If $p_k = p$,
  a) Obtain new plan $\mathbf{U}_p(k) = \mathbf{U}_p^{\mathrm{opt}}(k)$ as solution to $\mathbb{P}_p^{\mathrm{D}}\big(\mathbf{x}_p(k); \mathbf{Z}_p^*(k)\big)$.
  b) Transmit new plan to agents in $\mathcal{Q}_p$.
  Else renew current plan via (14): $\mathbf{U}_p(k) = \tilde{\mathbf{U}}_p(k)$.
(v) Go to step (ii).

This algorithm requires that a feasible initial plan – *i.e.*, part of a feasible solution to the initial centralized problem $\mathbb{P}^{\mathrm{C}}$ – be made available to each control agent, a common assumption of DMPC methods; for example, see Richards and How (2007), Dunbar (2007). Note that the constraints of $\mathbb{P}^{\mathrm{C}}$ are *not* sequence dependent, and therefore the set of feasible initial plans is not sequence dependent. In fact, given an initial solution to $\mathbb{P}^{\mathrm{C}}$, recursive feasibility holds for the system controlled by DMPC for any subsequent choice of sequence, as Theorem 3.1 shows.

A further requirement is that the terminal set $\mathcal{X}_{F_p}$ for the local optimization be made available centrally, since coupling constraints must be satisfied therein. However, note that no further centralized processing is required from that point on.

The following theorem states the main result of the paper.

Theorem 3.1: *Suppose the sequence* $\mathbf{U}_p^*(k_0) = \big\{\bar{\mathbf{x}}_p^*(k_0|k_0), \bar{\mathbf{u}}_p^*(k_0|k_0), \ldots, \bar{\mathbf{u}}_p^*(k_0+N-1|k_0)\big\}, \forall p \in \mathcal{P}$, *exists and is a feasible (but not necessarily optimal) solution to* $\mathbb{P}^{\mathrm{C}}\big(\mathbf{x}_1(k_0), \ldots, \mathbf{x}_{N_p}(k_0)\big)$ *at some time step* k_0. *Then, for all* $\mathbf{x}_p(k_0+1) \in \mathbf{A}_p\mathbf{x}_p(k_0) + \mathbf{B}_p\mathbf{u}_p(k_0) \oplus \mathcal{W}_p, \forall p \in \mathcal{P}$, *where* $\mathbf{u}_p(k_0) = \bar{\mathbf{u}}_p^*(k_0|k_0) + \mathbf{K}_p\left(\mathbf{x}_p(k_0) - \bar{\mathbf{x}}_p^*(k_0|k_0)\right)$, *(i) the candidate sequence* $\tilde{\mathbf{U}}_p(k_0+1)$, *defined by* (14), *is a feasible solution to* $\mathbb{P}_p^{\mathrm{D}}\big(\mathbf{x}_p(k_0+1); \mathbf{Z}_p^*(k_0+1)\big)$*; (ii) the upper bound on the local cost decreases monotonically:*

$$J_p^*\big(\mathbf{x}_p(k_0+1); \mathbf{Z}_p^*(k_0+1)\big) \leq J_p^*\big(\mathbf{x}_p(k_0); \mathbf{Z}_p^*(k_0)\big) - l_p\big(\bar{\mathbf{x}}_p^*(k_0|k_0), \bar{\mathbf{u}}_p^*(k_0|k_0)\big),$$

for all $p \in \mathcal{P}$, *where* $J_p^*\big(\mathbf{x}_p(k_0); \mathbf{Z}_p^*(k_0)\big) = J_p\big(\mathbf{U}_p^*(k_0)\big)$*; and (iii) subsequently, the resulting closed-loop system controlled by Algorithm* **1** *is robustly-feasible and stable for any choice of update sequence.*

Proof For (i), given a feasible solution $\big\{\mathbf{U}_p^*(k_0)\big\}_{p\in\mathcal{P}}$ to $\mathbb{P}^{\mathrm{C}}\big(\mathbf{x}_1(k_0), \ldots, \mathbf{x}_{N_p}(k_0)\big)$, by Mayne et al. (2005, Proposition 3), $\big\{\tilde{\mathbf{U}}_p(k_0+1)\big\}_{p\in\mathcal{P}}$ is a feasible solution to $\mathbb{P}^{\mathrm{C}}\big(\mathbf{x}_1(k_0+1), \ldots, \mathbf{x}_{N_p}(k_0+1)\big)$. $\tilde{\mathbf{U}}_p(k_0+1)$ is also a feasible solution to $\mathbb{P}_p^{\mathrm{D}}\big(\mathbf{x}_p(k_0+1); \mathbf{Z}_p^*(k_0+1)\big)$, for any p; Proposition 3 in Mayne et al. (2005) implies that local constraints (8a) to (8f) are directly satisfied, while constraint (16) is satisfied by the choice $\bar{\mathbf{z}}_{cp}(\cdot|k_0+1) = \bar{\mathbf{z}}_{cp}^*(\cdot|k_0), \forall c \in \mathcal{C}_p$, so that $\sum_{p\in\mathcal{P}_c} \bar{\mathbf{z}}_{cp}^*(k_0+j|k_0) \in \mathcal{Z}_c, j \in \big\{1, \ldots, N\big\}$. This is then equivalent to constraint (8g) in the problem $\mathbb{P}^{\mathrm{C}}\big(\mathbf{x}_1(k_0+1), \ldots, \mathbf{x}_{N_p}(k_0+1)\big)$, (all $c \notin \mathcal{C}_p, p \notin \mathcal{P}_c$, have $\bar{\mathbf{z}}_{cp} = \mathbf{0}$).

For (ii), the value of local cost associated with the feasible $\mathbf{U}_p^*(k_0)$ at time k_0 is $J_p^*\big(\mathbf{x}_p(k_0); \mathbf{Z}_p^*(k_0)\big) = J_p\big(\mathbf{U}_p^*(k_0)\big)$. Then at time k_0+1, all non-updating subsystems $p \neq p_{k_0+1}$ adopt their respective candidate solutions, $\mathbf{U}_p(k_0+1) = \tilde{\mathbf{U}}_p(k_0+1)$, defined by (14), with associated cost

$$\begin{aligned}
\tilde{J}_p\big(\mathbf{x}_p(k_0+1); \mathbf{Z}_p^*(k_0+1))\big) &= J_p\big(\tilde{\mathbf{U}}_p(k_0+1)\big) \\
&= J_p\big(\mathbf{U}_p^*(k_0)\big) - l_p\big(\bar{\mathbf{x}}_p^*(k_0|k_0), \bar{\mathbf{u}}_p^*(k_0|k_0)\big) \\
&\quad + l_p\Big(\bar{\mathbf{x}}_p^*(k_0+N|k_0), \kappa_{F_p}\left(\bar{\mathbf{x}}_p^*(k_0+N|k_0)\right)\Big) \\
&\quad + F_p\Big(\mathbf{A}_p\bar{\mathbf{x}}_p^*(k_0+N|k_0) + \mathbf{B}_p\kappa_{F_p}\left(\bar{\mathbf{x}}_p^*(k_0+N|k_0)\right)\Big) \\
&\quad - F_p\big(\bar{\mathbf{x}}_p^*(k_0+N|k_0)\big).
\end{aligned}$$

By (12), the latter three terms sum to less than or equal to zero, leaving

$$\tilde{J}_p\big(\mathbf{x}_p(k_0+1); \mathbf{Z}_p^*(k_0+1)\big) \leq J_p^*\big(\mathbf{x}_p(k_0); \mathbf{Z}_p^*(k_0)\big) - l_p\big(\bar{\mathbf{x}}_p^*(k_0|k_0), \bar{\mathbf{u}}_p^*(k_0|k_0)\big),$$

for all $p \neq p_{k_0+1}$.

The optimizing subsystem p_{k_0+1} obtains $\mathbf{U}_{p_{k_0+1}}(k_0+1)$ as the solution to the local optimization $\mathbb{P}_{p_{k_0+1}}^{\mathrm{D}}\big(\mathbf{x}_{p_{k_0+1}}(k_0+1); \mathbf{Z}_{p_{k_0+1}}^*(k_0+1)\big)$; as $\tilde{\mathbf{U}}_{p_{k_0+1}}(k_0+1)$ is a known feasible solution, then an upper bound on the optimal cost is obtained

$$J^{\text{opt}}_{p_{k_0+1}}\big(\mathbf{x}_{p_{k_0+1}}(k_0+1);\mathbf{Z}^*_{p_{k_0+1}}(k_0+1)\big) \le \tilde{J}_p\big(\mathbf{x}_p(k_0+1);\mathbf{Z}^*_p(k_0+1)\big).$$

Thus, for *any* subsystem $p \in \mathcal{P}$, it follows that

$$J^*_p\big(\mathbf{x}_p(k_0+1);\mathbf{Z}^*_p(k_0+1)\big) \le J^*_p\big(\mathbf{x}_p(k_0);\mathbf{Z}^*_p(k_0)\big) - l_p\big(\bar{\mathbf{x}}^*_p(k_0|k_0),\bar{\mathbf{u}}^*_p(k_0|k_0)\big),$$

where J^*_p is the cost of a general feasible solution.

Part (iii) follows by applying recursion to (i) and (ii). Firstly, by construction, any solution $\mathbf{U}^*_{p_k}(k)$ to $\mathbb{P}^{\text{D}}_{p_k}\big(\mathbf{x}_{p_k}(k);\mathbf{Z}^*_{p_k}(k)\big)$ taken with the candidate solutions $\big\{\tilde{\mathbf{U}}_p(k)\big\}, p \ne p_k$, is a solution to $\mathbb{P}^{\text{C}}\big(\mathbf{x}_1(k),\dots,\mathbf{x}_{N_p}(k)\big)$; solving $\mathbb{P}^{\text{D}}_{p_k}$ is equivalent to solving $\mathbb{P}^{\text{C}}$ with $p \ne p_k$ constrained to take $\mathbf{U}_p(k) = \tilde{\mathbf{U}}_p(k)$. A feasible solution to $\mathbb{P}^{\text{C}}\big(\mathbf{x}_1(0),\dots,\mathbf{x}_{N_p}(0)\big)$ then implies all subsequent optimizations $\mathbb{P}^{\text{D}}_p\big(\mathbf{x}_p(k);\mathbf{Z}^*_p(k)\big), k \ge 0$, are feasible, regardless of the choice of update sequence $\{p_k\}_k$. Next, because $J^*_p(k+1) - J^*_p(k) \le -l_p\big(\bar{\mathbf{x}}^*_p(k|k),\bar{\mathbf{x}}^*_p(k|k)\big)$, yet $J^*_p(\cdot)$ and the stage cost $l_p(\cdot,\cdot)$ are both strictly non-negative, then by recursion it follows that $J^*_p(k+1) - J^*_p(k) \to 0$ as $k \to \infty$. In turn, this implies that $l_p\big(\bar{\mathbf{x}}^*_p(k|k),\bar{\mathbf{u}}^*_p(k|k)\big) \to 0$. Because $l_p\big(\mathbf{x}_p,\mathbf{u}_p\big) \ge c\|\mathbf{x}_p,\mathbf{u}_p\|$ for some $c > 0$, and $l_p\big(\mathbf{0},\mathbf{0}\big) = 0$, it must be that the nominal state $\bar{\mathbf{x}}^*_p(k|k) \to \mathbf{0}$ and the nominal control $\bar{\mathbf{u}}^*_p \to \mathbf{0}$. Finally, by the fact that $\mathbf{x}_p(k) \in \bar{\mathbf{x}}_p(k|k) \oplus \mathcal{R}_p, \forall k$, it follows that the true state $\mathbf{x}_p(k) \to \mathcal{R}_p$ as $k \to \infty$, and, furthermore,

$$\begin{aligned}\mathbf{u}_p(k) &= \bar{\mathbf{u}}^*_p(k|k) + \mathbf{K}_p\big(\mathbf{x}_p(k) - \bar{\mathbf{x}}^*_p(k|k)\big)\\ &\to \mathbf{K}_p\mathbf{x}_p(k)\end{aligned}$$

as $k \to \infty$. □

4 Communication analysis

It remains to evaluate exactly what information, denoted $\mathbf{Z}^*_p(k)$, is required in the local optimization for p. In the problem $\mathbb{P}^{\text{D}}_p\big(\mathbf{x}_p(k);\mathbf{Z}^*_p(k)\big)$, the structure in the coupling constraints, identified in (3) and (4), has been exploited. Firstly, only constraints $c \in \mathcal{C}_p$ are applied, as by definition (4), $\bar{\mathbf{z}}_{cp}(k+j|k) = \mathbf{0}$ for all other constraints $c \notin \mathcal{C}_p$, so these outputs do not affect the update of subsystem p. Secondly, the summation in (16), for each c, includes output terms from only those subsystems in $\mathcal{P}_c$; by definition (3), $\bar{\mathbf{z}}_{cr}(k+j|k) = \mathbf{0}$ for all other subsystems $r \notin \mathcal{P}_c$. The coupling terms $\bar{\mathbf{z}}^*_{cq}(k+j|k), \forall q \in \mathcal{P}_c \setminus \{p\}$ are not affected by the decision variables $\mathbf{U}_p(k)$, so they appear as fixed values in (16), denoted by *. Using the definition of coupled subsystems (5), it follows that to evaluate (16), values for $\bar{\mathbf{z}}^*_{cq}(k+j|k), \forall c \in \mathcal{C}_p$, are required from all other subsystems q in $\mathcal{Q}_p$.

We note, therefore, that it is not necessary to obtain the whole plan $\mathbf{U}^*_q(k)$ from some coupled q. Instead, define a message vector from subsystem p regarding constraint c at time k as

$$\mathbf{m}_{cp}(k) \triangleq \big[\bar{\mathbf{z}}^*_{cp}(k|k)^{\text{T}} \dots \bar{\mathbf{z}}^*_{cp}(k+N-1|k)^{\text{T}}\ \bar{\mathbf{x}}^*_p(k+N|k)^{\text{T}}\big]^{\text{T}}, \tag{17}$$

which includes the coupling outputs and the terminal state. Again, the * superscript denotes a feasible solution. Also, define a propagation matrix,

$$\Pi_{cp} \triangleq \begin{bmatrix} \mathbf{0}\,\mathbf{I}\,\mathbf{0} \dots & \mathbf{0} \\ \mathbf{0}\,\mathbf{0}\,\mathbf{I} \dots & \mathbf{0} \\ \vdots\ \vdots\ \vdots\ \ddots & \vdots \\ \mathbf{0}\,\mathbf{0}\,\mathbf{0} \dots & (\mathbf{E}_{cp} + \mathbf{F}_{cp}\mathbf{K}_{F_p}) \\ \mathbf{0}\,\mathbf{0}\,\mathbf{0} \dots & (\mathbf{A}_p + \mathbf{B}_p\mathbf{K}_{F_p}) \end{bmatrix},$$

assuming a linear terminal control law, *i.e.*, $\kappa_{F_p}(\mathbf{x}_p) = \mathbf{K}_{F_p}\mathbf{x}_p$, so that $\mathbf{m}_{cp}(k) = \Pi_{cp}\mathbf{m}_{cp}(k-1)$ is the message at time k for a non-updating subsystem $p \neq p_k$. Suppose the last time a subsystem p optimized its plan was at a step $\hat{k}_p$, before the current step k, defined as

$$\hat{k}_p(k) \triangleq \max_{k' \in \{k' < k | p_{k'} = p\}} k'. \tag{18}$$

Then the message at k for a subsystem p that last optimized at $\hat{k}_p$ is $\mathbf{m}_{cp}(k) = \Pi_{cp}^{(k-\hat{k}_p)}\mathbf{m}_{cp}(\hat{k}_p)$. Relating this back to the information that is required by p_k to evaluate (16), $\mathbf{Z}^*_{p_k}(k)$ is obtained as

$$\begin{aligned} \mathbf{Z}^*_{p_k}(k) &= \left\{\mathcal{I}\mathbf{m}_{cq}(k)\right\}_{c \in \mathcal{C}_{p_k}, q \in \mathcal{Q}_{p_k}} \\ &= \left\{\mathcal{I}\Pi_{cq}^{(k-\hat{k}_q)}\mathbf{m}_{cq}(\hat{k}_q)\right\}_{c \in \mathcal{C}_{p_k}, q \in \mathcal{Q}_{p_k}}, \end{aligned} \tag{19}$$

where the matrix operator $\mathcal{I} \triangleq \mathrm{diag}(\mathbf{I}, \mathbf{I}, \dots, \mathbf{0})$ removes the terminal states. The inclusion of the terminal state $\bar{\mathbf{x}}_p(k+N|k)$ in the message permits the correct propagation for steps $k > \hat{k}_q + N$. This propagation leads to the following requirement for obtaining $\mathbf{Z}^*_{p_k}(k)$.

Requirement 4.1 At a time step k, the control agent for an optimizing subsystem p_k must have received messages $\mathbf{m}_{cq}(\hat{k}_q), \forall c \in \mathcal{C}_{p_k}$, from all subsystems $q \in \mathcal{Q}_{p_k}$.

This illustrates a key feature of tube MPC that means it lends itself to distribution; an updating subsystem p_k may obtain $\mathbf{Z}^*_{p_k}(k)$ by using Π_{cq} to propagate previously-communicated data regarding coupled subsystems, with no communication required in the interim. Therefore, to meet Requirement 4.1 it is sufficient for each agent p to transmit the message $\mathbf{m}_{cp}, \forall c \in \mathcal{C}_p$, to all $q \in \mathcal{Q}_p$ after each planning update, as in Algorithm **1**.

However, instances exist where message transmissions are *not* necessary. The remainder of this section identifies these instances, and shows how flexibility in update sequence choice can be exploited to offer a DMPC scheme with low levels of communication. The measure of communication that shall be used in the sequel is, with only small loss of generality, the number of data exchanges between any pair of subsystems at a time step. A data exchange occurs whenever a subsystem agent transmits its message to any other subsystem agent. This overlooks the fact that messages may be of different sizes. However, this approach is justified, since the "cost" of communication is often driven by connectivity rather than bandwidth.

It is observed that after the optimization at time k, the updating system p_k needs to transmit a message if both the following two criteria are met:

C1: The optimized plan differs from the candidate plan, *i.e.*, $\mathbf{U}^{\mathrm{opt}}_{p_k}(k) \neq \tilde{\mathbf{U}}_{p_k}(k)$;
C2: Before subsystem p_k next optimizes, another subsystem in $\mathcal{Q}_{p_k}$ will optimize.

Otherwise, the new information transmitted by p_k is redundant. It follows that, following an optimization, a subsystem p_k must transmit its plan to all others in $\mathcal{Q}_{p_k}$ if C1 and C2 are met.

Similarly, it is possible to establish the communication required for the centralized implementation of the controller (CMPC). If an optimization is to take place at time k, then a central agent must have received $\mathbf{x}_p(k)$ from *all* subsystems prior to the optimization. Susbequently, new plans must be communicated to all subsystems. Assuming that the control agent is located on one of the subsystems $p \in \mathcal{P}$, the minimum number of data exchanges required at an optimization is therefore $2(N_p - 1)$.

In the worst case, when coupling constraints exist between all subsystems, subsystem p_k is coupled to all other subsystems, and the number of coupled agents is $n(\mathcal{Q}_{p_k}) = (N_p - 1)$ for any p_k. By definition, $n(\mathcal{Q}_{p_k}) \leq (N_p - 1)$; thus, DMPC requires, at most, only half as many data exchanges per optimization as does CMPC. However, lower levels of communication can be obtained by exploiting the the coupling structure.

For centralized MPC, at each time step, a decision is made whether to optimize or not. The resulting number of data exchanges that take place over the length of a simulation is then inextricably linked to the number of updating steps. With the distributed algorithm, we have an extra degree of freedom, in that the decision is not only whether to optimize or not, but also which subsystem is to optimize. For example, the sequence $\{1, 2, 1, 2, \ldots\}$ requires communication at every step, whereas $\{1, 1, 2, 2, \ldots\}$ requires communication at alternating steps. There is a many-to-one mapping of update sequences to data exchanges; thus, the link between the number of updating steps and communication is broken. It remains to determine the effect this flexibility has on system-wide performance, and this is explored in the next section.

5 Numerical examples

This section presents simulation results using the new distributed MPC algorithm. The first example compares the performance of DMPC with that of CMPC by investigating the trade between performance and communication, and shows that the flexibility in communication can be exploited to obtain better performance for DMPC with low levels of communication. The second example investigates the effect of delays on the performance of the proposed DMPC.

In both examples, a comparison is made with the *constraint-tightening* (CT-DMPC) method of Richards and How (2007). That method shares certain similarities with the DMPC proposed in this paper: CT-DMPC also guarantees robust constraint satisfaction and feasibility for subsystems coupled through the constraints, by updating agents' plans sequentially. However, CT-DMPC uses a fixed, pre-determined sequence for updating plans, and – based on the assumption of instantaneous data exchanges – all agents optimize within the same time step.

Consider the system consisting of N_p identical point masses moving in 1-D, each with double integrator dynamics, discretized using a time step $T = 1$ second. For all $p \in \mathcal{P}$:

$$\mathbf{A}_p = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{B}_p = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}.$$

Each mass is subject to local constraints on speed and control, *i.e.*, $\left|[0 \quad 1]\mathbf{x}_p\right| \leq 2$ and $\left|u_p(k)\right| \leq 1$, and all pairs are coupled by a constraint to remain 'close': $\left|[1 \quad 0]\left(\mathbf{x}_p - \mathbf{x}_q\right)\right| \leq \Delta x, \forall p \neq q$. The feedback controller is chosen to be the nilpotent controller, $\mathbf{K}_p = \begin{bmatrix}-1 & -1.5\end{bmatrix}$, such that $\left(\mathbf{A}_p + \mathbf{B}_p\mathbf{K}_p\right)^2 = \mathbf{0}$. Then the sets $\mathcal{R}_p$ are finitely-determined, and given by $\mathcal{W}_p \oplus \left(\mathbf{A}_p + \mathbf{B}_p\mathbf{K}_p\right)\mathcal{W}_p$, where $\mathcal{W}_p$ in this case is a simple hypercube, $\left\{\mathbf{w}_p \in \mathbb{R}^2 : \|\mathbf{w}_p\|_\infty \leq 0.1\right\}$.

The objective function is the quadratic form

$$l_p\left(\mathbf{x}_p, \mathbf{u}_p\right) = \mathbf{x}_p^{\mathrm{T}}\mathbf{Q}\mathbf{x}_p + \mathbf{u}_p^{\mathrm{T}}\mathbf{R}\mathbf{u}_p,$$
$$F_p\left(\mathbf{x}_p\right) = \mathbf{x}_p^{\mathrm{T}}\mathbf{P}\mathbf{x}_p,$$

where $\mathbf{Q} = \mathbf{I}_2$, $\mathbf{R} = 0.01$, and $\mathbf{P} = \begin{bmatrix} 2.0066 & 0.5099 \\ 0.5099 & 1.2682 \end{bmatrix}$ is the terminal cost matrix associated with the optimal, nominal, unconstrained LQR problem $\left(\mathbf{A}_p, \mathbf{B}_p, \mathbf{Q}, \mathbf{R}\right)$. The terminal control law is chosen as the LQR controller, *i.e.*, $\kappa_{F_p} = \mathbf{K}^{\mathrm{LQR}} = \begin{bmatrix}-0.6609 & -1.3261\end{bmatrix}$. Subsequently, the terminal sets $\mathcal{X}_{F_p}$ for the distributed algorithm are the maximal output-admissible invariant sets (Kolmanovsky and Gilbert 1998) associated with this control, in which coupling constraints are satisfied in a decoupled manner; *i.e.*, $x_{p,1} \leq 0.5\Delta x$ for each p. On the other hand, the centralized algorithm is provided with a larger, *centralized* version of this set, in which coupling constraint satisfaction is achieved in a centralized – rather than decoupled – sense.

Note that the same controllers and terminal set are also used for the CT-DMPC implementation, permitting a fair comparison.

5.1 *Performance versus communication*

A number of simulations were performed, varying the number of subsystems, the update sequence and the maximum separation distance, Δx. The initial states were $\mathbf{x}_p(0) = [20 \quad 0]^{\mathrm{T}}, \forall p \in \mathcal{P}$, with a horizon of 20 steps, and each mass subject to a random disturbance sequence throughout a simulation. The update sequence was varied in a different manner for CMPC, DMPC and CT-DMPC. For CMPC, a simple *mark-space* scheme was employed, where a mark represents an updating step and a space represents a zero-update step. The resulting sequence is repeated periodically to form the update sequence for the simulation. For example, for a mark value of 3 and a space value of 2, the resulting sequence is $\{c, c, c, 0, 0, c, c, c, 0, 0, \ldots\}$, where c denotes a centralized optimization.

For DMPC, a similar mark-space scheme is used, but with an additional degree of freedom. It is assumed that the subsystems optimize in a cyclical manner. Then, n_1 denotes the number of repetitions of update steps per subsystem (marks), n_2 denotes the number of zero-update steps (spaces), and n_3 denotes the number of extra zero-update steps that follow the completion of a cycle. For example, with $n_1 = 2, n_2 = 3, n_3 = 4$:

$$\{c, \underbrace{1, 1}_{n_1}, \underbrace{0, 0, 0}_{n_2}, 2, 2, 0, 0, 0, \ldots, \underbrace{N_p, N_p}_{n_1}, \underbrace{0, 0, 0}_{n_2}, \underbrace{0, 0, 0, 0}_{n_3}, \ldots\},$$

where c denotes the initial centralized step.

Finally, the update sequence for CT-DMPC was chosen to resemble to centralized sequence, but where a mark step corresponds to all agents updating in the preset sequence $\{1, 2, \ldots, N_p\}$. This amounts to employing the algorithm in its originally-intended, sequential manner (Richards and How 2007), yet permitting the communication levels to vary by introducing zero-update steps where all agents adopt the candidate plans. Each algorithm is initialized with an optimal centralized plan at $k = 0$.

Figure 2 shows plots of closed-loop cost against communication, in which a 'good' controller is one whose data point lies close to the bottom left of the graph. Results are shown as the convex hulls of points obtained for each controller by varying the update sequence, and as (i) the number of subsystems varies (left to right), (ii) the separation distance Δx increases (top to bottom). The measure of performance in this instance is the value of the stage cost, summed over the duration of the simulation and all subsystems. As discussed in Section 4, the measure of communication is the number of data exchanges between subsystems. As expected, all the graphs for both DMPC and CMPC show a trade: better performance can be achieved by using more communication.

Firstly, on the comparison between the centralized and distributed forms of tube MPC, in the majority of cases, the plots show regions where the closed-loop objective values for tube DMPC are *lower* than the corresponding CMPC values for the same level of communication. Predictably, at very high levels of communication, CMPC performs better than DMPC. This is intuitive since DMPC solves the same optimization but in a more constrained manner. However, at low levels of communication, DMPC can perform better. This is enabled by the extra degree of freedom in the DMPC update sequence, breaking the link between computation and communication levels. In tube DMPC, it is possible to construct an update sequence in which some subsystem replans at every step, but communication is required far less frequently. Furthermore, the range of communication for which DMPC outperforms CMPC can be seen to increase as either Δx increases or N_p decreases. These movements correspond to making the optimization less tightly coupled, thus giving more flexibility for local decision making.

Comparing these results to those obtained for CT-DMPC, that method obtains – in the majority of cases – better performance at all levels of communication. Furthermore, in most cases performance is better than for even centralized tube MPC. Although communication for CT-DMPC can scale poorly as the number of subsystems increases, even then instances exist where performance at low communication levels is better than any tube MPC implementation. In fact, the CT method for robustness is less conservative than the tube method (Trodden 2009). However, and crucially, the CT-DMPC algorithm relies on instantaneous inter-agent transfers of data during a time step, while none of the tube DMPC exchanges require this. The effect that *delays* in this inter-agent communication have on performance is studied in the next example.

5.2 *Effect of delays*

Two different delays were introduced to the problem: $D_{\text{comp}} < T$ is the time delay between a local agent's measuring of its state and the subsequent updating of its control input (following optimization) during a time step of length T, while $D_{\text{comm}} < T$ is the time taken to successfully communicate a new plan to other agents. For CT-DMPC, it is assumed that the pth agent may not optimize during step k until information is received from agent $p-1$ from earlier in the same interval (Richards and How 2007). Conversely, tube DMPC allows the whole interval $[k + D_{\text{comp}}, k+1)$ for communication.

For a two-mass system, with $\mathbf{x}_1(0) = [5 \quad 1]^{\mathrm{T}}$ and $\mathbf{x}_2(0) = [5 \quad 0]^{\mathrm{T}}$, the delays D_{comp} and D_{comm} were varied over the intervals $[0, 0.25T]$ and $[0, 0.5T]$ respectively, where $T = 1$ second. All parameters are the same as in the previous section, with the exception of the horizon, which was shortened to $N = 7$ to reflect the closer proximity to the origin of the initial states. During each simulation, disturbances were applied to force the masses apart: $\mathbf{w}_1(k) = [1 \quad 1]^{\mathrm{T}}$ and $\mathbf{w}_2(k) = -[1 \quad 1]^{\mathrm{T}}$ for all k.

Figure 3 compares the values of closed-loop cost obtained for both tube DMPC and CT-DMPC. As shown by the previous example, where no delays are present CT-DMPC achieves best performance. As delays are lengthened, the cost values for CT-DMPC are seen to increase, more severely so for D_{comp}. Where cost data are absent over the delay domain, the system violated the constraints; CT-DMPC goes infeasible for $\frac{D_{\text{comp}}}{T} \gtrsim 0.175$ and additionally for high total delay $D_{\text{comp}} + D_{\text{comm}}$. On the other hand, the system controlled by tube DMPC achieves robust feasibility over the whole domain. In addition, although higher than those of CT-DMPC for low values of delay, the cost values increase approximately linearly with D_{comp} and do not increase with D_{comm}. Consequently, tube DMPC out-performs CT-DMPC when delays are longer. This result confirms that CT-DMPC – with its reliance on instantaneous data exchanges – is the more susceptible of the two methods to the effect of delays, both in terms of feasibility and performance. Furthermore, it highlights a key feature of the tube DMPC method: that at least the full remainder of one time step is available for information exchange following an optimization.

6 Conclusions

In this paper, a formulation has been presented for robust distributed model predictive control of LTI subsystems coupled through the constraints. The order of optimization for each subsystem is unrestricted and communication between subsystems is required only when relevant updates are performed, leading to flexible communications. The new formulation extends the tube MPC concept to a distributed implementation and inherits its property of robust feasibility and stability despite persistent disturbances. By exploiting the greater communication flexibility of the new algorithm, better performance can be achieved than centralized tube MPC when

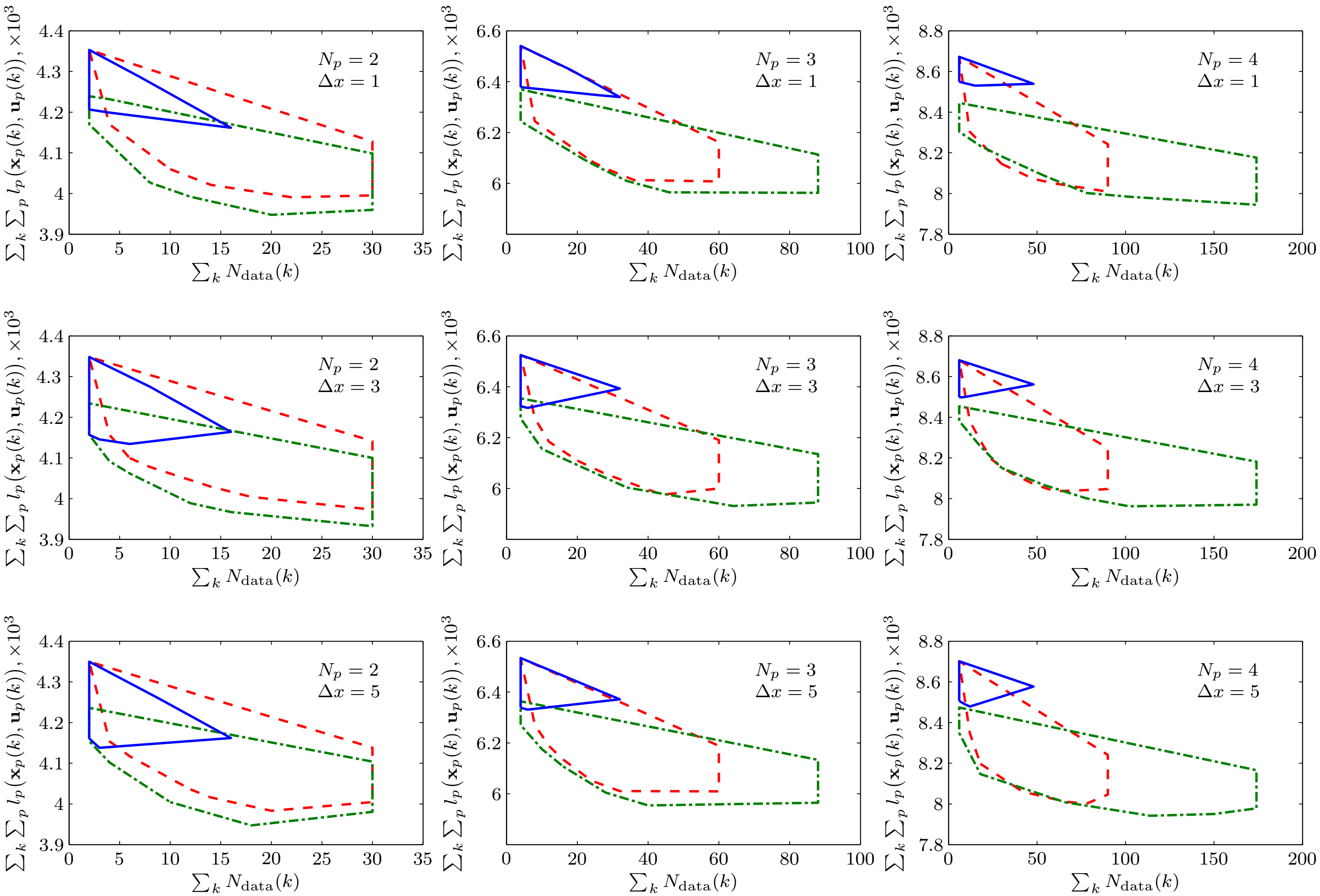

Figure 2. Closed-loop cost for DMPC (solid), tube CMPC (dash), and CT-DMPC (dash-dot). The convex hulls of points, obtained by varying update sequences, are shown.

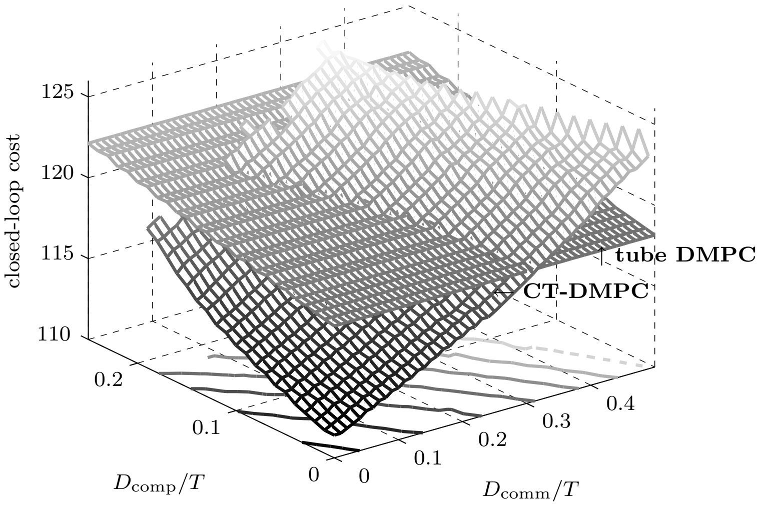

Figure 3. Surfaces of closed-loop cost versus communication delay D_{comm} and computation delay D_{comp}. Contours are additionally shown for CT-DMPC.

communication is limited. Furthermore, a comparison with a similar robust distributed method has shown that the new algorithm offers a clear benefit, in terms of feasibility and performance, when computational and communication delays are present.

On-going research is investigating how to obtain closed-loop performance given a particular structure of coupling constraints. In particular, inter-agent cooperation may be employed – by including a consideration of other subsystems' objectives in the local cost function – to promote system-wide performance by avoiding 'greedy' local decision-making.

References

Alessio, A., and Bemporad, A. (2007), "Decentralized Model Predictive Control of Constrained Linear Systems," in *Proceedings of the European Control Conference*, Jul, EUCA, pp. 2813–2818.

Borrelli, F., Keviczky, T., and Stewart, G.E. (2005), "Decentralized Constrained Optimal Control Approach to Distributed Paper Machine Control," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, Dec, pp. 3037–3042.

Camponogara, E., Jia, D., Krogh, B.H., and Talukdar, S. (2002), "Distributed Model Predictive Control," *IEEE Control Systems Magazine*, 22, 44–52.

Du, X., Xi, Y., and Li, S. (2001), "Distributed Model Predictive Control for Large-Scale Systems," in *Proceedings of the American Control Conference*, Jun, pp. 3142–3143.

Dunbar, W.B. (2007), "Distributed Receding Horizon Control of Dynamically Coupled Nonlinear Systems," *IEEE Transactions on Automatic Control*, 52, 1249–1263.

Dunbar, W.B., and Murray, R.M. (2006), "Distributed Receding Horizon Control for Multi-Vehicle Formation Stabilization," *Automatica*, 42, 549–558.

Franco, E., Magni, L., Parisini, T., Polycarpou, M.M., and Scattolini, R. (2008), "Cooperative Constrained Control of Distributed Agents with Nonlinear Dynamics and Delayed Information Exchange," *IEEE Transactions on Automatic Control*, 53, 324–338.

Franco, E., Parisini, T., and Polycarpou, M.M. (2007), "Design and stability analysis of cooperative receding-horizon control of linear discrete-time agents," *International Journal of Robust and Nonlinear Control*, 17, 982–1001.

Fukushima, H., and Bitmead, R.R. (2005), "Robust constrained predictive control using comparison model," *Automatica*, 41, 97–106.

Giovanini, L., Balderud, J., and Katebi, M.R. (2007), "Autonomous and decentralized mission planning for clusters of UUVs," *International Journal of Control*, 80, 1169–1179.

Jia, D., and Krogh, B.H. (2002), "Min-Max Feedback Model Predictive Control For Distributed Control With Communication," in *Proceedings of the American Control Conference*, May, pp. 4507–4512.

Keviczky, T., Borrelli, F., and Balas, G.J. (2006), "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, 42, 2105–2115.

Kim, T.H., and Sugie, T. (2005), "Robust Decentralized MPC Algorithm for a class of Dynamically Interconnected Constrained Systems," in *Proceedings of the IEEE Conference on Decision and Control*.

Kolmanovsky, I., and Gilbert, E.G. (1998), "Theory and Computation of Disturbance Invariant Sets for Discrete-Time Linear Systems," *Mathematical Problems in Engineering*, 4, 317–367.

Kuwata, Y., Richards, A., Schouwenaars, T., and How, J. (2007), "Distributed Robust Receding Horizon Control for Multivehicle Guidance," *IEEE Transactions on Control Systems Technology*, 14, 627–641.

Ling, K.V., Maciejowski, J.M., and Wu, B. (2005), "Multiplexed model predictive control," in *16th IFAC World Congress*, Jul, Prague.

Maciejowski, J.M., *Predictive Control with Constraints*, Prentice Hall (2002).

Magni, L., and Scattolini, R. (2006), "Stabilizing decentralized model predictive control of nonlinear systems," *Automatica*, 42, 1231–1236.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M. (2000), "Constrained model predictive control: Stability and optimality," *Automatica*, 36, 789–814.

Mayne, D.Q., Seron, M.M., and Raković, S.V. (2005), "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, 41, 219–224.

Raffard, R.L., Tomlin, C.J., and Boyd, S.P. (2004), "Distributed Optimization for Cooperative Agents: Application to Formation Flight," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, pp. 2453–2459.

Richards, A., and How, J.P. (2005), "Implementation of Robust Decentralized Model Predictive Control," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Aug, American Institute of Aeronautics and Astronautics.

Richards, A., and How, J.P. (2007), "Robust Distributed Model Predictive Control," *International Journal of Control*, 80, 1517–1531.

Scattolini, R. (2009), "Architectures for distributed and hierarchical Model Predictive Control — A review," *Journal of Process Control*, 19, 723–731.

Scokaert, P.O.M., and Mayne, D.Q. (1998), "MinMax Feedback Model Predictive Control for Constrained Linear Systems," *IEEE Transactions on Automatic Control*, 43, 1136–1142.

Shim, D.H., Kim, H.J., and Sastry, S. (2003), "Decentralized Reflective Model Predictive Control of Multiple Flying Robots in Dynamic Environment," in *Proceedings of the 42nd IEEE Conference on Decision and Control*.

Trodden, P.A. (2009), "Robust Distributed Control of Constrained Linear Systems," PhD dissertation, University of Bristol.

Venkat, A.N., Rawlings, J.B., and Wright, S.J. (2004), "Plant-Wide Optimal Control With Decentralized MPC," in *Proceedings of the 7th IFAC International Symposium on Dynamics and Control of Process Systems*, Jul.

Venkat, A.N., Hiskens, I.A., Rawlings, J.B., and Wright, S.J. (2008), "Distributed MPC Strategies With Application to Power System Automatic Generation Control," *IEEE Transactions on Control Systems Technology*, 16, 1192–1206.

Waslander, S.L., Inalhan, G., and Tomlin, C.J. (2004), "Decentralized Optimization via Nash Bargaining," (Vol. 4, eds. D. Grundel, R. Murphey and P.M. Pardolos, World Scientific, chap. 1, pp. 1–21.