Chapter 3
Cooperative Tube-based Distributed MPC for Linear Uncertain Systems Coupled Via Constraints

P. A. Trodden and A. G. Richards

Abstract This chapter presents a robust form of distributed model predictive control for multiple, dynamically decoupled subsystems subject to bounded, persistent disturbances. Control agents make decisions locally and exchange plans; satisfaction of coupling constraints is ensured by permitting only non-coupled subsystems to update simultaneously. Robustness to disturbances is achieved by use of the tube MPC concept, in which a local control agent designs a tube, rather than a trajectory, for its subsystem to follow. Cooperation between agents is promoted by a local agent, in its optimization, designing hypothetical tubes for other subsystems, and trading local performance for global. Uniquely, robust feasibility and stability are maintained without the need for negotiation or bargaining between agents.

3.1 Introduction

This chapter presents a distributed form of MPC for systems defined by the following characteristics: the overall system is composed of, or may be decomposed to, a number of dynamically decoupled subsystems. Each has linear, time-invariant dynamics, and is subject to local constraints and persistent, bounded disturbances. The subsystems are coupled via constraints, and should coordinate decision-making to satisfy these constraints robustly and also to minimize some system-wide cost.

In the described approach, the distributed control agents exchange plans to achieve constraint satisfaction. Key features are that (i) coupled subsystems may not update

P. A. Trodden (✉)
Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK
e-mail: p.trodden@sheffield.ac.uk

A. G. Richards
Department of Aerospace Engineering, University of Bristol, Bristol, UK
e-mail: arthur.richards@bris.ac.uk

J. M. Maestre and R. R. Negenborn (eds.), *Distributed Model Predictive Control Made Easy*, Intelligent Systems, Control and Automation: Science and Engineering 69,
DOI: 10.1007/978-94-007-7006-5_3,

their plans simultaneously; (ii) robust stability is guaranteed for any choice of update sequence; (iii) an agent communicates only when strictly necessary, and (iv) cooperation between agents is promoted by a local agent considering the objectives of, and designing hypothetical plans for, other subsystems. The resulting algorithm offers flexibility in communication and computation, and requires no inter-agent negotiation, iteration or bargaining.

The approach, which first appeared in its non-cooperative form [10], uses the concept of tube MPC [6], a form of robust MPC that guarantees feasibility and stability despite the action of an unknown but bounded disturbance. The approach shares similarities with the ‘sequential’ DMPC method of Richards and How [7], in that robust feasibility and stability of the overall system is guaranteed by local agents updating plans one at a time, without negotiation. However, tube DMPC permits a flexible order of updating, in contrast to a fixed, pre-determined sequence. Thus, this approach combines guaranteed robust feasibility and convergence, in the presence of a persistent disturbance, with flexible communication.

In the cooperative form of the algorithm [11], a local agent designs not only its own tube, but also hypothetical tubes for other subsystems in the problem. The idea is that an agent may now consider the objectives and intentions of others in order to arrive at a more cooperative solution. Here, cooperation is taken to mean the improvement of system-wide performance through the avoidance of greedy behaviour by individual agents. Coupled constraint satisfaction is, however, maintained without the need for inter-agent negotiation or bargaining. In comparison, approaches to cooperation based on inter-agent iteration or bargaining [4, 8, 15], require multiple and repeated information exchanges at each time step in order to achieve constraint satisfaction and stability. Thus, the approach combines robust satisfaction of coupled constraints with cooperation. Overall, cooperation offers performance close to that of centralized MPC but with less computation and communication.

The chapter begins with a formal statement of the problem. In Sect. 3.3, the distributed MPC approach is described, including local optimization problems, algorithms and communication requirements. Theoretical results are summarized in Sect. 3.4. Finally, Sect. 3.5 discusses applications of the approach.

3.2 Problem Statement

The system under consideration consists of a set $\mathcal{N}$ of dynamically decoupled subsystems. Subsystem dynamics are linear and time invariant (LTI):

$$\mathbf{x}_i(k+1) = \mathbf{A}_i \mathbf{x}_i(k) + \mathbf{B}_i \mathbf{u}_i(k) + \mathbf{d}_i(k), \quad \forall i \in \mathcal{N}, k \in \mathbb{N}.$$

For a subsystem i, $\mathbf{x}_i \in \mathbb{R}^{n_{x_i}}$, $\mathbf{u}_i \in \mathbb{R}^{n_{u_i}}$ and $\mathbf{d}_i \in \mathbb{R}^{n_{x_i}}$ are, respectively, the state, control input and disturbance. The latter is unknown *a priori*, but is assumed to lie in a known independent, bounded, compact set that contains the origin in its interior:

$$\mathbf{d}_i(k) \in \mathcal{D}_i \subset \mathbb{R}^{n_{x_i}}, \quad \forall i \in \mathcal{N}, k \in \mathbb{N}.$$

Each subsystem $i \in \mathcal{N}$ is subject to local constraints on an output $\mathbf{y}_i \in \mathbb{R}^{n_{y_i}}$:

$$\mathbf{y}_i(k) = \mathbf{C}_i\mathbf{x}_i(k) + \mathbf{D}_i\mathbf{u}_i(k) \in \mathcal{Y}_i,$$

where $\mathcal{Y}_i \subset \mathbb{R}^{n_{y_i}}$ is closed and contains the origin in its interior. The subsystems are coupled via constraints, constructed as follows. Define a coupling output $\mathbf{z}_{ci} \in \mathbb{R}^{n_{z_c}}$ for a constraint $c \in \mathcal{C}$ and a subsystem $i \in \mathcal{N}$. The sum of coupling outputs for a constraint c must lie in a closed set $\mathcal{Z}_c$ that contains the origin:

$$\mathbf{z}_{ci}(k) = \mathbf{E}_{ci}\mathbf{x}_i(k) + \mathbf{F}_{ci}\mathbf{u}_i(k), \text{ and} \sum_{i \in \mathcal{N}} \mathbf{z}_{ci}(k) \in \mathcal{Z}_c.$$

The following definitions identify structure in these coupling constraints, and are used later in the requirements for communication. Let $\mathcal{N}_c$ be the set of subsystems involved in constraint c, and $\mathcal{C}_i$ the set of constraints involving subsystem i:

$$\mathcal{N}_c \triangleq \{i \in \mathcal{N} : [\mathbf{E}_{ci}\ \mathbf{F}_{ci}] \neq \mathbf{0}\}, \tag{3.1}$$

$$\mathcal{C}_i \triangleq \{c \in \mathcal{C} : [\mathbf{E}_{ci}\ \mathbf{F}_{ci}] \neq \mathbf{0}\}. \tag{3.2}$$

Then the set of all other subsystems coupled to a subsystem i is

$$\mathcal{Q}_i = \left(\bigcup_{c \in \mathcal{C}_i} \mathcal{N}_c\right) \setminus \{i\}. \tag{3.3}$$

The control objective is, without loss of generality, to steer each subsystem state $\mathbf{x}_i$ to the origin, while satisfying constraints. To this end, assume that each $(\mathbf{A}_i, \mathbf{B}_i)$ is controllable, and that the state $\mathbf{x}_i$ is available to the control agent for subsystem i at each sampling instant. Define $\mathbf{K}_i \in \mathbb{R}^{n_{u_i} \times n_{x_i}}$ as a stabilizing controller for each $i \in \mathcal{N}$, let $\mathcal{R}_i$ be a disturbance-invariant set [3] for the resulting controlled system. That is, $(\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_i)\mathbf{x}_i + \mathbf{d}_i \in \mathcal{R}_i$ for all $\mathbf{x}_i \in \mathcal{R}_i$ and $\mathbf{d}_i \in \mathcal{D}_i$; equivalently, $(\mathbf{A}_i + \mathbf{B}_i\mathbf{K}_i)\mathcal{R}_i \oplus \mathcal{D}_i \subseteq \mathcal{R}_i$. It is assumed that the disturbance sets $\mathcal{D}_i$ are sufficiently small such that $(\mathbf{C}_i + \mathbf{D}_i\mathbf{K}_i)\mathcal{R}_i \subset \text{interior}(\mathcal{Y}_i)$, $\forall i \in \mathcal{N}$, and $\bigoplus_{i \in \mathcal{N}}(\mathbf{E}_{ci} + \mathbf{F}_{ci}\mathbf{K}_{ci})\mathcal{R}_i \subset \text{interior}(\mathcal{Z}_c)$, $\forall c \in \mathcal{C}$. This latter assumption is not unusual and represents a mild condition for many practical constraints and disturbances.

3.3 Distributed MPC Using Tubes

Tube MPC was introduced by Mayne et al. in [6]. Instead of optimizing a sequence of states, i.e. points in the state space, it optimizes for a *tube*, or a sequence of sets, within the state space. Drawing on invariance concepts, one can prove that once in the tube, the system state can stay in the tube despite the disturbance. A consequence of this is that the initial centre of the tube is variable. One can envisage this as the problem of catching a speck of dust with a vacuum cleaner: the centre of the hose can be placed anywhere, provided the speck is within the hose opening. The internal dynamics of the hose will then get the dust to the bag, without having to move the hose further. It is this invariance that makes the tube approach so attractive for DMPC: if each subsystem has a tube that is invariant under local control, the tube does not need to be updated to accommodate the disturbance. Hence, communication is necessary only when an agent chooses to change the tube. Coupling can be captured by ensuring that the tubes are consistent across the system.

This section begins with a review of centralized tube MPC, also serving to introduce the relevant notation. Then two different forms of the distributed approach are described: one that leads to a non-cooperative form of DMPC, and a second that uses a conceptual extension to promote cooperation between agents.

3.3.1 Review of Centralized Tube MPC

The tube MPC approach [6] uses the nominal system dynamics to design a sequence of disturbance-invariant state sets. The decision variable includes not only the control sequence for each subsystem i, $\bar{\mathbf{u}}_i(k : k+N_p-1) \triangleq \left[\bar{\mathbf{u}}_i^{\mathrm{T}}(k), \bar{\mathbf{u}}_i^{\mathrm{T}}(k+1), \ldots, \bar{\mathbf{u}}_i^{\mathrm{T}}(k+N-1)\right]^{\mathrm{T}}$, but also the initial state predictions, $\bar{\mathbf{x}}_i(k)$ for all i, which correspond to the tube centres. As the optimization involves only nominal terms, complexity is comparable to standard MPC. Robustness to disturbances is guaranteed by use of a feedback law to keep the state of each subsystem around its tube centre.

For the system state $\left\{\mathbf{x}_i(k)\right\}_{i\in\mathcal{N}}$, the centralized optimal control problem is

$$\min_{\{\bar{\mathbf{x}}_i(k),\bar{\mathbf{u}}_i(k:k+N_p-1)\}_{i\in\mathcal{N}}} \sum_{i\in\mathcal{N}} J_i\left(\bar{\mathbf{x}}_i(k), \bar{\mathbf{u}}_i(k : k+N_p-1)\right) \tag{3.4}$$

subject to, $\forall l \in \{1, \ldots, N_p - 1\}, i \in \mathcal{N}$,

$$\bar{\mathbf{x}}_i(k+l+1) = \mathbf{A}_i\bar{\mathbf{x}}_i(k+l) + \mathbf{B}_i\bar{\mathbf{u}}_i(k+l), \tag{3.5a}$$

$$\bar{\mathbf{y}}_i(k+l) = \mathbf{C}_i\bar{\mathbf{x}}_i(k+l) + \mathbf{D}_i\bar{\mathbf{u}}_i(k+l), \tag{3.5b}$$

$$\bar{\mathbf{z}}_{ci}(k+l) = \mathbf{E}_{ci}\bar{\mathbf{x}}_i(k+l) + \mathbf{F}_{ci}\bar{\mathbf{u}}_i(k+l), \forall c \in \mathcal{C}, \tag{3.5c}$$

$$\mathbf{x}_i(k) - \bar{\mathbf{x}}_i(k) \in \mathcal{R}_i, \tag{3.5d}$$

$$\bar{\mathbf{x}}_i(k+N_p) \in \mathcal{X}_i^{\mathrm{f}}, \tag{3.5e}$$

$$\bar{\mathbf{y}}_i(k+l) \in \tilde{\mathcal{Y}}_i, \tag{3.5f}$$

$$\sum_{i\in\mathcal{N}} \bar{\mathbf{z}}_{ci}(k+l) \in \tilde{\mathcal{Z}}_c, \forall c \in \mathcal{C}. \tag{3.5g}$$

Here, the local cost function for each subsystem is

$$J_i\left(\bar{\mathbf{x}}_i(k), \bar{\mathbf{u}}_i(k:k+N_p-1)\right) = F_i\left(\bar{\mathbf{x}}_i(k+N)\right) + \sum_{l=0}^{N_p-1} L_i\left(\bar{\mathbf{x}}_i(k+l), \bar{\mathbf{u}}_i(k+l)\right), \quad (3.6)$$

where the stage cost $L_i : \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{n_{u_i}} \mapsto \mathbb{R}_{0+}$, and $F_i : \mathbb{R}^{n_{x_p}} \mapsto \mathbb{R}_{0+}$ is a terminal cost.

The constraint sets in the problem, $\tilde{\mathcal{Y}}_i$, $\tilde{\mathcal{Z}}_c$ are tightened versions of the original sets, in order to provide a margin for uncertainty:

$$\tilde{\mathcal{Y}}_i = \mathcal{Y}_i \sim (\mathbf{C}_i + \mathbf{D}_i\mathbf{K}_i)\mathcal{R}_i, \forall i \in \mathcal{N}, \quad (3.7a)$$

$$\tilde{\mathcal{Z}}_c = \mathcal{Z}_i \sim \bigoplus_{i \in \mathcal{N}} (\mathbf{E}_{ci} + \mathbf{F}_{ci}\mathbf{K}_i)\mathcal{R}_i, \forall c \in \mathcal{C}. \quad (3.7b)$$

The sets $\mathcal{R}_i$ are the cross-sections of the tubes, and satisfy the assumptions in the previous section; the tube itself is given by $\left\{\bar{\mathbf{x}}_i(k) \oplus \mathcal{R}_i, \ldots, \bar{\mathbf{x}}_i(k+N_p) \oplus \mathcal{R}_i\right\}$ for a subsystem i. The tightening is necessary to accommodate the tube approach: instead of dealing with the exact output values, the constraints act upon the centres of the tubes, and the actual outputs could be anywhere within the cross-section.

The sets $\mathcal{X}_i^{\mathrm{f}}$ for all $i \in \mathcal{N}$ are terminal constraint sets, and each is assumed to be an admissible control invariant set [3]. That is, there is assumed to exist a control law $\mathbf{u}_i = \kappa_i^{\mathrm{f}}(\mathbf{x}_i)$ such that, for all $\mathbf{x}_i \in \mathcal{X}_i^{\mathrm{f}}$,

$$\mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\kappa_i^{\mathrm{f}}(\mathbf{x}_i) \in \mathcal{X}_i^{\mathrm{f}}, \quad (3.8a)$$

$$\mathbf{C}_i\mathbf{x}_i + \mathbf{D}_i\kappa_i^{\mathrm{f}}(\mathbf{x}_i) \in \tilde{\mathcal{Y}}_i, \quad (3.8b)$$

$$\sum_{i \in \mathcal{N}} \mathbf{E}_{ci}\mathbf{x}_i + \mathbf{F}_{ci}\kappa_i^{\mathrm{f}}(\mathbf{x}_i) \in \tilde{\mathcal{Z}}_c, \quad \forall c \in \mathcal{C}. \quad (3.8c)$$

It is often assumed that $\kappa_i^{\mathrm{f}}(\mathbf{x}_i) = \mathbf{K}_i\mathbf{x}_i$, but this is not necessary, and an alternative choice of κ_i^{f} may simplify the determination of the sets $\mathcal{X}_i^{\mathrm{f}}$. Here, we leave the problem in its most flexible form, and the reader is referred to the examples cited in Sect. 3.5 for further options.

Assuming that a feasible solution, $\left\{\bar{\mathbf{x}}_i^*(k), \bar{\mathbf{u}}_i^*(k:k+N_p-1)\right\}_{i \in \mathcal{N}}$, is available to this problem at time k, the following control is applied to each subsystem

$$\mathbf{u}_i(k) = \bar{\mathbf{u}}_i^*(k) + \mathbf{K}_i\left(\mathbf{x}_i(k) - \bar{\mathbf{x}}_i^*(k)\right). \quad (3.9)$$

The resulting controlled system, under the assumptions described, is recursively feasible, despite the actions of the persistent disturbances [6]. This is because, given a feasible solution $\left\{\bar{\mathbf{x}}_i^*(k), \bar{\mathbf{u}}_i^*(k:k+N_p-1)\right\}_{i \in \mathcal{N}}$ for time k, the candidate or "tail" solution $\left\{\bar{\mathbf{x}}_i^*(k+1), \bar{\mathbf{u}}_i^*(k+1:k+N_p)\right\}_{i \in \mathcal{N}}$, where

$$\bar{\mathbf{x}}_i^*(k+1) = \mathbf{A}_i \bar{\mathbf{x}}_i^*(k) + \mathbf{B}_i \bar{\mathbf{u}}_i^*(k), \tag{3.10a}$$

$$\bar{\mathbf{u}}_i^*(k+1:k+N_p) = \left[\bar{\mathbf{u}}_i^{*\mathrm{T}}(k+1), \ldots, \bar{\mathbf{u}}_i^{*\mathrm{T}}(k+N_p-1), \kappa_i^{\mathrm{f}}(\bar{\mathbf{x}}_i^*(k+N_p))^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{3.10b}$$

is a feasible solution to the centralized problem at $k+1$. Furthermore, with standard assumptions [5] on the stage and terminal costs, asymptotic or exponential convergence of the states of the system to the sets $\mathcal{R}_i$ is guaranteed [6].

In the sequel, let $\mathbf{U}_i(k) = \{\bar{\mathbf{x}}_i(k), \bar{\mathbf{u}}_i(k:k+N_p-1)\}$. $\mathbf{U}_i^*(k)$ is a feasible solution for step k, and $\tilde{\mathbf{U}}_i(k+1)$ denotes the candidate solution for $k+1$, as formed by (3.10).

3.3.2 Distributed Tube MPC

In the distributed MPC approach described in the remainder of this section, the centralized problem is distributed among the subsystem control agents as local optimization problems. In order to maintain coupled constraint satisfaction, only a subset of agents solve their optimizations for a new plan. Meanwhile, the other agents 'freeze' their plans by adopting the tail solution (3.10) unchanged.

In the non-cooperative form, a control agent for subsystem i minimizes only its local share of the system-wide cost (3.4), which is J_i, as defined by (3.6). The local optimization problem, $P_i^{\mathrm{dmpc}}(\mathbf{x}_i(k), \mathbf{Z}_i^*(k))$, for i is

$$\min_{\mathbf{U}_i(k)} J_i(\mathbf{U}_i(k)) \tag{3.11}$$

subject to local constraints (3.5a)–(3.5f) for i, and the coupling constraint

$$\bar{\mathbf{z}}_{ci}(k+l) + \sum_{j \in \mathcal{N}_c \setminus \{i\}} \bar{\mathbf{z}}_{cj}^*(k+l) \in \tilde{\mathcal{Z}}_c, \forall l \in \{1, \ldots, N_p-1\}, c \in \mathcal{C}_i. \tag{3.12}$$

where * denotes a fixed, previously published output of a coupled subsystem. $\mathbf{Z}_i^*(k)$ denotes the aggregate information required by i to evaluate the coupling constraints.

The optimization is employed in Algorithm 3.1, to be executed by all agents in parallel.

Though the algorithm is executed by all agents in parallel, only agents in a set $\mathcal{N}_k \subseteq \mathcal{N}$ are permitted to update by optimization at a step k. All other agents $j \notin \mathcal{N}_k$ renew their current plans, as per (3.10), each by shifting in time the tail of its previous, feasible plan and augmenting with a step of terminal control.

Having obtained a plan $\mathbf{U}^*(k)$ at step k, by either optimization or renewal, each agent unilaterally applies the first control of the planned sequence. No negotiation or iterative refinement of solutions takes place during a time step.

The order in which the subsystems' plans are optimized is determined by the *update sequence*, $\{\mathcal{N}_1, \ldots, \mathcal{N}_k, \mathcal{N}_{k+1}, \ldots\}$. This is to be chosen by the designer,

Algorithm 3.1 Distributed MPC for a subsystem i

1: Wait for an initial feasible plan $\mathbf{U}_i^*(0) = \left\{\bar{\mathbf{x}}_i^*(0), \bar{\mathbf{u}}_i^*(0 : N_p - 1)\right\}$ and information, including $\mathbf{Z}_i^*(0)$. Set $k = 0$.
2: Apply control $\mathbf{u}_i(k) = \bar{\mathbf{u}}_i^*(k) + \mathbf{K}_i\left(\mathbf{x}_i(k) - \bar{\mathbf{x}}_i^*(k)\right)$. Wait one time step, increment k.
3: Measure state $\mathbf{x}_i(k)$.
4: Update plan. If $i \in \mathcal{N}_k$,

1. Propagate intentions of other agents, $\bar{\mathbf{z}}_{cj}^*(\cdot)$, to the current planning horizon.
2. Obtain new plan $\mathbf{U}_i^{\text{opt}}(k)$ by solving the local problem $P_i^{\text{dmpc}}(k)$.
3. Set $\mathbf{U}_i^*(k) = \mathbf{U}_i^{\text{opt}}(k)$.
4. If necessary, transmit new information to other agents.

Else, renew existing plan: form $\tilde{\mathbf{U}}_i(k)$ according to (3.10) and set $\mathbf{U}_i^*(k) = \tilde{\mathbf{U}}_i(k)$.
5: Go to step 2.

and may be a static (*i.e.,* pre-determined) or dynamic sequence. For a particular time step k, the only criterion for the selection of the updating set of agents, $\mathcal{N}_k$, is that no two optimizing subsystems may share coupled constraints. That is, $(i, j) \in \mathcal{N}_k$ only if $j \notin \mathcal{Q}_i$. In the limiting case of coupling between all pairs of subsystems, this reduces the maximum size of $\mathcal{N}_k$ to a single agent. Note, in addition, that the empty set is always a valid choice for $\mathcal{N}_k$, such that no optimization is solved at a time k.

3.3.2.1 Initialization of DMPC

It is assumed that all of the local subsystem information required to formulate the local problem is made available off-line to agent i.

Requirement 3.1 *[Local subsystem information] The matrices* $\mathbf{A}_i$, $\mathbf{B}_i$, $\mathbf{C}_i$, $\mathbf{D}_i$, $\mathbf{E}_{ci}$, $\mathbf{F}_{ci}$, $\forall c \in \mathcal{C}_i$, *stabilizing controllers* $\mathbf{K}_i$ *and* κ_i^{f}, *sets* $\mathcal{R}_i$, $\mathcal{X}_i^{\text{f}}$, $\tilde{\mathcal{Y}}_i$, $\tilde{\mathcal{Z}}_c$, $\forall c \in \mathcal{C}_i$, *and cost function* J_i *shall be available to the control agent for subsystem* i.

The computation of $\mathbf{K}_i$, κ_i^{f}, $\mathcal{R}_i$ and $\mathcal{X}_i^{\text{f}}$—as well as the tightening of sets according to (3.7)—need be done only once, off-line, at initialization. It is assumed that this is done centrally, with the results communicated to agents.

Also, since each agent needs to be able to deduce its neighbors' current intentions based on their last published plans, information is required on the dynamics and constraints of those neighbours:

Requirement 3.2 *[Coupled subsystem information] The matrices* $\mathbf{A}_j$, $\mathbf{B}_j$, $\mathbf{E}_{cj}$, $\mathbf{F}_{cj}$, $\forall c \in \mathcal{C}_i$ *and terminal controller* κ_i^{f} *shall be available to the control agent for subsystem i for each coupled subsystem* $j \in \mathcal{Q}_i$.

Finally, it is required that a feasible initial plan is available to each control agent.

Requirement 3.3 *[Initial plan] A feasible local plan* $\mathbf{U}_i^*(0)$ *shall be available to the control agent for subsystem i at time* $k = 0$.

This is a common assumption of many DMPC approaches (e.g. [1, 7]). Note that this does not imply that the centralized problem must be solved to optimality; often a simple feasible solution is available, such as all subsystems remaining stationary. No further centralized processing is required after this initialization step.

3.3.2.2 Inter-agent Communication

An updating agent $i \in \mathcal{N}_k$ must have received sufficient information from each coupled agent $j \in \mathcal{Q}_i$ so as to enforce constraints (3.12). Specifically, this requires the construction of signals $\bar{\mathbf{z}}^*_{cj}(k+l)\ \forall l \in \{1, \ldots, N_p - 1\}, \forall j \in \mathcal{N}_c, c \in \mathcal{C}_i$.

It is not necessary to obtain the whole *plan* $\mathbf{U}^*_j(k)$ from some coupled $j \in \mathcal{Q}_i$. Instead, define a *message* vector for subsystem i regarding constraint c at time k as

$$\mathbf{m}_{ci}(k) \triangleq \left[\bar{\mathbf{z}}^{*\mathrm{T}}_{ci}(k) \ldots \bar{\mathbf{z}}^{*\mathrm{T}}_{ci}(k+N_p-1)\ \bar{\mathbf{x}}^{*\mathrm{T}}_{i}(k+N_p)\right]^{\mathrm{T}}, \tag{3.13}$$

which includes the terminal state, and where * denotes a feasible solution.

Put simply, each updating agent must have the latest information available about every other coupled agent. More formally:

Requirement 3.4 *[Information exchange] Consider any two coupled agents i and $j \in \mathcal{Q}_i$ and any two time steps k_i and $k_j > k_i$ such that $i \in \mathcal{N}_{k_i}$ and $j \in \mathcal{N}_{k_j}$, i.e. agents i and j updated at times k_i and k_j respectively. Then message $\mathbf{m}_{ci}(k)$ for every coupling constraint $c \in \mathcal{C}_i \bigcap \mathcal{C}_j$ must have been sent from i to j at least once during time steps $k \in [k_i, k_j]$.*

A sufficient means of achieving this is for an agent $i \in \mathcal{N}_k$, following update, to transmit $\mathbf{m}_{ci}(k)$ regarding constraints $c \in \mathcal{C}_i \bigcap \mathcal{C}_j$ to each agent $j \in \mathcal{Q}_j$.

Note that information in $\mathbf{m}_{cj}(k)$ may be “out of date” if sent at an earlier time, in the sense that it may not include outputs for all times in the current planning horizon. However, since it is required that coupled agents must adopt their tail plans unless publishing otherwise, plans for others can be brought up to date by the propagation step 4.1 in Algorithm 3.1. The knowledge of others’ dynamics from Requirement 3.2 and the inclusion of the terminal state $\bar{\mathbf{x}}^*_i(k+N_p)$ in the message $\mathbf{m}_{cj}(k)$ ensures that this can be done.

Minimal communication strategies vary significantly with different update sequences. The reader is referred to [14] for a full formal coverage of both communication and propagation mathematics.

3.3.3 Cooperative DMPC

A shortcoming of the method described in the previous section, and of many DMPC methods, is that ‘greedy’ local decision making can lead to poor system-wide performance [15]. Even if the dynamics and objectives are decoupled, the closed-loop performance of subsystems coupled via the constraints is coupled. Consequently, the solutions applied by agents can be severely sub-optimal—even with inter-agent iteration—and hence cooperation is required to obtain good performance.

A logical approach to promoting cooperation is for distributed control agents to consider, in addition to their own objectives, the objectives or intentions of other agents in the system. To help illustrate this meaning of cooperation, consider an analogy of driving in a long congested stream of traffic. From our car, we observe another car waiting to turn on to our road. Considering only our local objective, we wish to get to our destination as quickly as possible, and we are not constrained to give way to the waiting car, so we continue. However, we could consider the objective of the waiting car as well as our own. A small sacrifice of our objective—the time to let the waiting car pull out—saves a long wait for the other car, and hence an improvement of the global objective. We don't need to instruct the other car to pull out though: merely by slowing to create the opportunity, it is natural for them to take it. This is the key to cooperation: considering the objectives of others can improve global performance without increasing communication requirements. Cooperative control could also be thought of as "considerate control".

This approach to cooperation has been shown to work well for subsystems coupled only via dynamics or objectives. However, it is generally incompatible with maintaining coupled constraint satisfaction. For example, in the method of [2] a local control agent designs, in addition to its own plan, *hypothetical* plans for directly-coupled subsystems. That is, the local problem for subsystem i is

$$\min_{\mathbf{U}_i(k), \hat{\mathbf{U}}^i_j(k), j \in \mathcal{Q}_i} J_i\big(\mathbf{U}_i(k)\big) + \sum_{j \in \mathcal{Q}_i} J_j\big(\hat{\mathbf{U}}^i_j(k)\big)$$

subject to (3.5a)–(3.5f) on $\mathbf{U}_i(k)$, similar local constraints on each $\hat{\mathbf{U}}^i_j(k)$, $j \in \mathcal{Q}_i$, and the coupling constraints

$$\bar{\mathbf{z}}_{ci}(k+l) + \sum_{j \in \mathcal{N}_c \setminus \{i\}} \hat{\mathbf{z}}^i_{cj}(k+l), \quad \forall c \in \mathcal{C}_i, l \in \{1, \ldots, N_p - 1\},$$

where $\{\cdot\}^i_j$ denotes a variable for an agent j that has been computed by agent i.

A crucial detail is that various representations of a plan for a subsystem i might exist at any instance. Firstly, a subsystem $j \in \mathcal{N}$ has the plan $\mathbf{U}^*_j(k)$ for time k, which it is currently following. In addition, an $i \in \mathcal{Q}_j$ has, as part of its own decision-making process, designed a hypothetical plan $\hat{\mathbf{U}}^i_j(k)$ for j that is not necessarily equal to $\mathbf{U}^*_j(k)$. In doing so, it has ensured satisfaction of the coupling constraints by its optimized, local plan $\mathbf{U}^{\text{opt}}_i(k)$ when taken together with the plans $\hat{\mathbf{U}}^i_j(k)$, $j \in \mathcal{Q}_i$. However, satisfaction of the constraints by $\mathbf{U}^{\text{opt}}_i(k)$ together with the actual plans $\mathbf{U}^*_j(k)$, $j \in \mathcal{Q}_i$—and therefore feasibility of the overall, closed-loop system—is not assured, even if only a single agent updates at a time step.

In the approach described in this section, a local agent i designs hypothetical plans for others, yet now this set of other subsystems is an arbitrary *cooperating set* $\mathcal{S}_i(k)$. The local problem is to minimize a weighted sum of local costs by designing a local plan $\mathbf{U}_i(k)$ and a hypothetical plan $\hat{\mathbf{U}}^i_j(k)$ for each $j \in \mathcal{S}_i(k)$. The problem is solved subject to local constraints on $\mathbf{U}_i(k)$ and each $\hat{\mathbf{U}}^i_j(k)$, and coupling constraints on $\mathbf{U}_i(k)$ together with (i) fixed $\mathbf{U}^*_j(k)$ for all coupled $j \in \mathcal{Q}_i$ and (ii) the hypothetical plans $\hat{\mathbf{U}}^i_j(k)$ for all $j \in \mathcal{S}_i$ and the fixed plans $\mathbf{U}^*_m(k)$ for all coupled $m \in \mathcal{Q}_i \setminus \mathcal{S}_i$. The additional decision variables $\{\hat{\mathbf{U}}^i_j(k)\}$ are internal to agent i's decision making and will not be communicated to other agents. Following the optimization, i communicates information about only its own plan, $\mathbf{U}^{\text{opt}}_i(k)$, as before. Moreover, there is no obligation for a cooperating subsystem $j \in \mathcal{S}_i$ to itself optimize at the next step or indeed ever adopt the plan $\hat{\mathbf{U}}^i_j(k)$. The main point is that the an agent i, in determining its own plan, considers what others may be able to achieve.

The presence of *two* sets of coupling constraints in the optimization is crucial in the development here. Effectively, two different representations of a plan for a cooperating subsystem $j \in \mathcal{S}_i$ appear in the local optimization for i: firstly, a previously published plan, $\mathbf{U}^*_j(k)$, originating from the last time step at which j optimized, and the plan that subsystem is currently following; secondly, a hypothetical plan, $\hat{\mathbf{U}}^i_j(k)$, designed locally by agent i. This leads to a key feature of the method; that of promoting inter-agent cooperation yet maintaining robust feasibility of all local decisions.

The local optimization problem, $P^{\text{cdmpc}}_i\left(\mathbf{x}_i(k), \breve{\mathbf{Z}}^*_i(k)\right)$, where $\breve{\mathbf{Z}}^*_i(k)$ denotes the extended information required, for subsystem i is formally defined as

$$\min_{\mathbf{U}_i(k), \hat{\mathbf{U}}^i_j(k), j \in \mathcal{S}_i} J_i\left(\mathbf{U}_i(k)\right) + \sum_{j \in \mathcal{S}_i(k)} \alpha^i_j J_j\left(\hat{\mathbf{U}}^i_j(k)\right) \tag{3.14}$$

subject to local constraints (3.5a)–(3.5f) on $\mathbf{U}_i(k)$, (3.5a)–(3.5c), (3.5e), (3.5f) on $\hat{\mathbf{U}}^i_j(k), \forall j \in \mathcal{S}_i(k)$, the initial constraints

$$\hat{\mathbf{x}}^i_j(k) = \bar{\mathbf{x}}^*_j(k), \tag{3.15a}$$

$$\hat{\mathbf{u}}^i_j(k) = \bar{\mathbf{u}}^*_j(k), \tag{3.15b}$$

for all $j \in \mathcal{S}_i$, and, for prediction steps $l \in \{1, \ldots, N_p - 1\}$, the coupling constraints

$$\bar{\mathbf{z}}_{ci}(k+l) + \sum_{j \in \mathcal{N}_c \setminus \{i\}} \bar{\mathbf{z}}^*_{cj}(k+l) \in \tilde{\mathcal{Z}}_c, \forall c \in \mathcal{C}_i, \tag{3.15c}$$

$$\bar{\mathbf{z}}_{ci}(k+l) + \sum_{j \in \mathcal{S}_i(k)} \hat{\mathbf{z}}^i_{cj}(k+l) + \sum_{m \in \mathcal{N}_c \setminus \{i, \mathcal{S}_i(k)\}} \bar{\mathbf{z}}^*_{cm}(k+l) \in \tilde{\mathcal{Z}}_c, \forall c \in \mathcal{C}_{\mathcal{S}_i(k)} \triangleq \bigcup_{j \in \mathcal{S}_i(k)} \mathcal{C}_j, \tag{3.15d}$$

Algorithm 3.2 Cooperative distributed MPC for a subsystem i

1: Wait for an initial feasible plan $\mathbf{U}_i^*(0) = \{\bar{\mathbf{x}}_i^*(0), \bar{\mathbf{u}}_i^*(0 : N_p - 1)\}$ and information, including $\check{\mathbf{Z}}_i^*(0)$. Set $k = 0$.
2: Apply control $\mathbf{u}_i(k) = \bar{\mathbf{u}}_i^*(k) + \mathbf{K}_i(\mathbf{x}_i(k) - \bar{\mathbf{x}}_i^*(k))$. Wait one time step, increment k.
3: Measure state $\mathbf{x}_i(k)$.
4: Update plan. If $i \in \mathcal{N}_k$,

   1. Choose cooperating set $\mathcal{S}_i(k)$ and weights α_j^i for each $j \in \mathcal{S}_i(k)$.
   2. Obtain new plan $\mathbf{U}_i^{\text{opt}}(k)$ by solving the local problem $P_i^{\text{cdmpc}}(\mathbf{x}_i(k), \check{\mathbf{Z}}_i^*(k))$.
   3. Set $\mathbf{U}_i^*(k) = \mathbf{U}_i^{\text{opt}}(k)$.
   4. Transmit new information to other agents.

   Else, renew existing plan: form $\tilde{\mathbf{U}}_i(k)$ according to (3.10) and set $\mathbf{U}_i^*(k) = \tilde{\mathbf{U}}_i(k)$.
5: Go to step 2.

The initial constraints (3.15a) and (3.15b), which replace (3.15d), provide the starting point of the hypothetical trajectory $\hat{\mathbf{U}}_j^i(k)$ for each $j \in \mathcal{S}_i(k)$. These constraints act on the assumption that any $j \in \mathcal{S}_i(k)$ can not optimize its own plan until, at the earliest, the next step $k+1$. Hence, these predicted trajectories may begin to diverge from the previously published trajectories, $\mathbf{U}_j^*(k)$, only at the $k+1$ prediction time.

The cooperative problem is solved in the Algorithm 3.2.

The cooperating set $\mathcal{S}_i(k)$ and the scalar weightings α_j^i are essentially tuning parameters for the level of cooperation. The parameter $\alpha_j^i \geq 0$ is the weighting applied to the local subsystem cost J_j for $j \in \mathcal{S}_i(k)$; smaller values ($\alpha_j^i < 1$) place more emphasis on i's own objective and self interest, while larger values ($\alpha_j^i > 1$) have the opposite effect. The size of the cooperating set maps to what portion of the system-wide cost is considered in the local optimization. If $\mathcal{S}_i(k)$ is empty, the problem reverts simply to the non-cooperative problem P_i^{dmpc}. Conversely, as $\mathcal{S}_i(k) \to \mathcal{N} \setminus \{i\}$, the local optimization attempts to solve a problem more closely resembling the system-wide, centralized problem, but with modified constraints.

As before, having obtained a plan $\mathbf{U}^*(k)$ at step k, by either optimization or renewal, each agent unilaterally applies the first control of the planned sequence. No negotiation or iterative refinement of solutions takes place during a time step, and hypothetical plans are never exchanged or compared.

3.3.3.1 Initialization of Cooperative DMPC

As before, it is assumed that each agent has the information required to formulate its local optimization problem (Requirements 3.1 and 3.2). It is also assumed that initial plans are available for each subsystem (Requirement 3.3).

3.3.3.2 Inter-agent Communication

The additional coupling constraints (3.15d) and initial intention constraints (3.15a) and (3.15b) both demand information beyond that required for non-cooperative DMPC.

Considering first the coupling constraints, constraint (3.15c) matches the non-cooperative counterpart (3.12), and hence Requirement 3.4 still applies. For the new constraint (3.15d), an updating agent further requires coupling information from any agents coupled to those in its cooperating set $\mathcal{S}_i(k)$. Formally:

Requirement 3.5 *[Additional coupling information exchange] Consider any three agents* i, j *and* h *such that* $i \in \mathcal{Q}_h$ *and* $h \in S_j$ *and any two time steps* k_i *and* $k_j > k_i$ *such that* $i \in \mathcal{N}_{k_i}$ *and* $j \in \mathcal{N}_{k_j}$*, i.e. agents* i *and* j *updated at times* k_i *and* k_j *respectively. Then message* $\mathbf{m}_{ci}(k)$ *for every coupling constraint* $c \in \mathcal{C}_i \bigcap \mathcal{C}_h$ *must have been sent from* i *to* j *at least once during time steps* $k \in [k_i, k_j]$.

For the initial intent constraints (3.15a) and (3.15b), full state and control outputs are required but might not be included in the coupling messages. Thus the requirement is for the latest complete plan information from any agents in the cooperating set:

Requirement 3.6 *[Cooperating plan exchange] Consider an updating agent* j *and another agent* $i \in \mathcal{S}_j$*, (*i.e. *agent* j *wants to cooperate with* i*) and any two time steps* k_i *and* $k_j > k_i$ *such that* $i \in \mathcal{N}_{k_i}$ *and* $j \in \mathcal{N}_{k_j}$*, i.e. agent* i *updated at time* k_i *and agent* j *updated at* k_j*. Then the plan* $\mathbf{U}_i^*(k)$ *must have been sent from* i *to* j *at least once during time steps* $k \in [k_i, k_j]$.

A sufficient, yet conservative, means of meeting these requirements is for the communication step in Algorithm 3.2 to specify transmission of the full plan to all other subsystems following update. While this may seem significant, it should be noted that to meet the requirement it is sufficient for one agent to transmit its plan to others only after that plan has changed, i.e., as a result of optimization. Moreover, it is not necessary for an agent to update at every time step, and robust coupled constraint satisfaction and stability are guaranteed for any choices of update sequence and cooperating sets. Thus, data exchanges need not occur at every time step, and the cooperating set and update sequence may be tailored to exploit this flexibility, as has been shown in [14] for the non-cooperative form.

3.4 Theoretical Results

3.4.1 Robust Constraint Satisfaction and Feasibility

With no assumptions extra to those already stated, the system controlled according to Algorithm 3.1 or 3.2 attains the properties of guaranteed robust constraint satisfaction and robust feasibility. This result, established in [14], relies on the observation that, given a feasible solution at time k_0, the candidate solution for each subsystem at the next time step $k_0 + 1$—as defined by (3.10)—is feasible for all possible realizations of the disturbances, $\mathbf{d}_i \in \mathcal{D}_i$, and any choice of update sequence.

In [11], this result is extended to the cooperative form of the algorithm by noting that, at k_0, an updating local agent i has available both a feasible local plan $\tilde{\mathbf{U}}_i(k_0)$ (*i.e.,* the candidate plan) and feasible hypothetical plans $\hat{\mathbf{U}}^i_j(k_0) = \tilde{\mathbf{U}}_j(k_0)$ for $j \in \mathcal{S}_i(k_0)$, and for any choice of cooperating set $\mathcal{S}_i(k_0)$. Viewed differently, the cooperative form of DMPC is equivalent to the non-cooperative form with only a modified cost, which happens to include the evaluation of options for other agents. Since constraint satisfaction and feasibility depend only on the constraints of each optimization and not on its cost, it is logical that cooperative DMPC inherits the feasibility properties of its non-cooperative counterpart.

3.4.2 Robust Convergence and Stability

Under further assumptions, asymptotic convergence of the states of the system to a neighbourhood of the origin is guaranteed.

(A1) The stage cost $L_i(\mathbf{x}_i, \mathbf{u}_i) \geq c\left\|(\mathbf{x}_i, \mathbf{u}_i)\right\|$, for $c > 0$, and $L_i(\mathbf{0}, \mathbf{0}) = 0$, $\forall i \in \mathcal{N}$.

(A2) The terminal cost is a local Lyapunov function: for all $\mathbf{x}_i \in \mathcal{X}^{\mathrm{f}}_i$ and $i \in \mathcal{N}$, $F_i\left(\mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\kappa^{\mathrm{f}}_i(\mathbf{x}_i)\right) - F_i\left(\mathbf{x}_i\right) \leq -L_i\left(\mathbf{x}_i, \kappa^{\mathrm{f}}_i(\mathbf{x}_i)\right)$.

(A3) The local cost of an adopted plan $\mathbf{U}^*_i(k)$ for any $i \in \mathcal{N}_k$ updating at k satisfies $J_i\left(\mathbf{U}^*_i(k)\right) \leq J_i\left(\tilde{\mathbf{U}}_i(k)\right) + \sum_{j\in\mathcal{N}} \epsilon_j L_j\left(\bar{\mathbf{x}}^*_j(k-1), \bar{\mathbf{u}}^*_j(k-1)\right)$ for some chosen $0 \leq \epsilon_j < 1, \forall j \in \mathcal{N}$, where $\tilde{\mathbf{U}}_i(k)$ is the candidate plan (3.10).

Together with the assumptions on $\mathbf{K}_i$, κ^{f}_i, $\mathcal{R}_i$, and $\mathcal{X}^{\mathrm{f}}_i$, Assumptions (A1) and (A2) represent a specific case of the standard assumptions (A1)–(A4) in [5].

The non-cooperative approach requires only Assumptions (A1) and (A2) [14]. It may be established that each $\mathbf{x}_i(k) \to \mathcal{R}_i$ and $\mathbf{u}_i \to \mathbf{K}_i\mathbf{u}_i$ as $k \to \infty$. This holds for all realizations of the disturbances $\mathbf{d}_i \in \mathcal{D}_i$, and for any choice of update sequence.

Robust convergence and stability of the cooperative form is established with the help of Assumption (A3). This limits the amount by which the local cost J_i of an agent's solution is permitted to *increase* over that of the candidate plan in order to benefit other agents. Intuitively, an unbounded increase may lead to instability if repeated by many agents over time. However, if (A3) holds, monotonic descent of the *global* cost, $\sum_{i\in\mathcal{N}} J_i\left(\mathbf{U}^*_i(k)\right)$ is assured, and each $\mathbf{x}_i(k) \to \mathcal{R}_i$ and $\mathbf{u}_i \to \mathbf{K}_i\mathbf{u}_i$ as $k \to \infty$. This holds regardless of disturbances, choice of update sequence, and choices of cooperating sets. Although difficult to prove, (A3) often holds anyway. Since it can be shown that it is always possible to satisfy (A3), it may be enforced by a direct constraint, though resulting in a more complex and constrained problem.

3.4.3 The Benefit of Cooperation

The problem of how the cooperating sets $\mathcal{S}_i$ are chosen to obtain the maximal benefit to system-wide performance is studied in [13]. It is proven that, depending on the coupling structure, it is not necessary to cooperate with all others in the problem, yet not sufficient to cooperative only with directly-coupled subsystems. An adaptive form of cooperation between agents is proposed, in which agents cooperate with others connected by *paths* in a graph of active coupling constraints

In [9], the DMPC approach is studied in a game-theoretical framework. Under assumptions milder than those required for asymptotic convergence of the states to a neighbourhood of the origin, (i.e., (A1) and (A2)), the states of the controlled system converge to some limit set. The system is in such a limit set if and only if the control agents are continually playing Nash solutions. Relating the Nash solutions to the cooperation set choices, it may be proven that increasing inter-agent cooperation does not enlarge the set of Nash solutions. Thence, it follows that increasing the size of cooperating sets does not enlarge the set of state limit sets for the system.

3.5 Applications of the Approach

Figure 3.1 shows results from an example, taken from [10], concerning the control of a group of point masses, coupled by the requirement to stay close together. The trade between communication and computation is shown for both centralized and tube DMPC. Centralized cannot reduce communication without lowering its update rate. Tube DMPC can manipulate the update sequence to change communication, and hence can out-perform centralized at low communication levels.

Figure 3.2 illustrates the benefits of cooperation when applied to a system of five point masses. The objective is for the masses to reach the origin, yet the coupled constraint on positions, $\sum_{i \in \mathcal{N}} x_{i,1} \geq 1$, means that this cannot be achieved by all masses simultaneously. Cooperative DMPC clearly delivers a fairer outcome.

In [13], the coupling structure is exploited, with the cooperating set for the point-mass system being chosen on-line, according to the active coupling constraints.

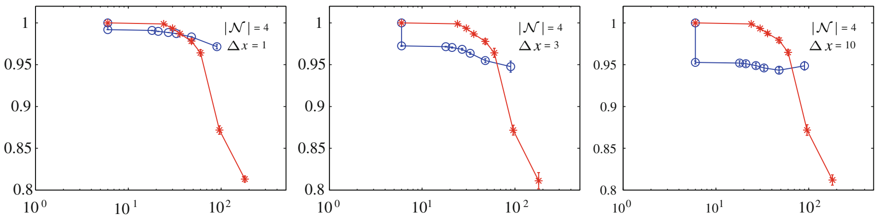


Fig. 3.1 Cost versus number of data exchanges for (i) DMPC (∘) and (ii) CMPC (∗). Each mass is required to remain within Δx of the others, with (l to r) $\Delta x = 1$, 3, and 10. Reproduced from [10]

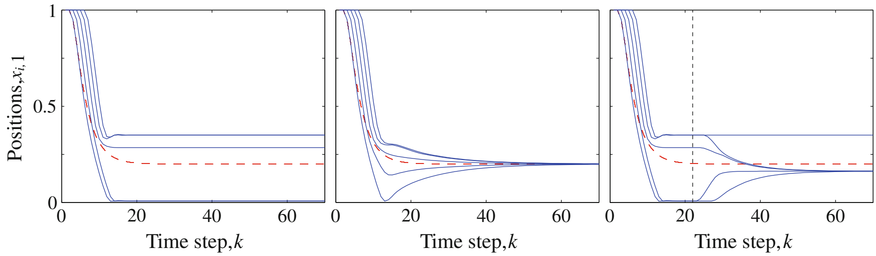
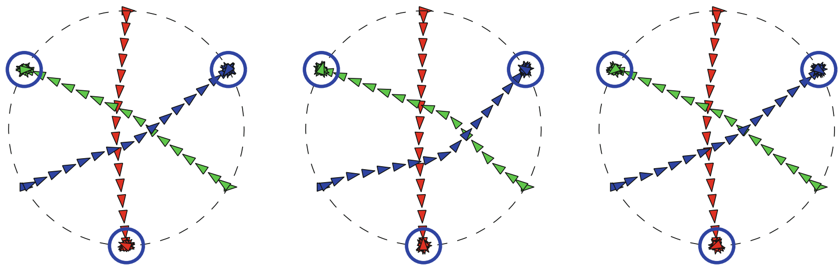


Fig. 3.2 Convergence of a five-mass system when controlled by (*left* to *right*) non-cooperative DMPC; cooperative DMPC with one other agent in the cooperating set $\mathcal{S}_i$; firstly non-cooperative DMPC, and then cooperative DMPC. Reproduced from [13]



(a) **(b)** **(c)**

Fig. 3.3 Three vehicles traversing a circle when controlled by **a** centralized MPC, **b** non-cooperative DMPC and **c** cooperative DMPC with the next-to-plan vehicle in the cooperating set for vehicle i. Reproduced from [11]

Simulations show that it is beneficial for an agent to cooperate with non-directly coupled agents, but not necessary to include all other masses in the cooperating set.

The distributed MPC algorithms described in this chapter are applicable to those systems comprising dynamically decoupled, LTI subsystems that share coupling constraints. A natural application is guidance and control of multiple vehicles, for which collision avoidance can be enforced via coupled constraints. In [11], a problem similar to air traffic control is simulated, and it is demonstrated that the cooperative approach leads to a more equitable arrangement of flight paths (Fig. 3.3). In [12], multiple vehicles are given the shared objective of achieving complete coverage or search of an area in minimum time.

References

1. W.B. Dunbar, Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Autom. Control* **52**(7), 1249–1263 (2007)
2. T. Keviczky, F. Borrelli, G.J. Balas, Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica* **42**(12), 2105–2115 (2006)

3. I. Kolmanovsky, E.G. Gilbert, Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering* **4**, 317–367 (1998)
4. Y. Kuwata, J.P. How, Cooperative distributed robust trajectory optimization using receding horizon MILP. *IEEE Transactions on Control Systems Technology* **19**(2), 423–431 (2011)
5. D.Q. Mayne, J.B. Rawlings, C.V. Rao, P.O.M. Scokaert, Constrained model predictive control: Stability and optimality. *Automatica* **36**, 789–814 (2000)
6. D.Q. Mayne, M.M. Seron, S.V. Raković, Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* **41**(2), 219–224 (2005)
7. A.G. Richards, J.P. How, Robust distributed model predictive control. *International Journal of Control* **80**(9), 1517–1531 (2007)
8. B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, G. Pannocchia, Cooperative distributed model predictive control. *Systems & Control Letters* **59**, 460–469 (2010)
9. P. A. Trodden, D. Nicholson, A. G. Richards, Distributed model predictive control as a game with coupled constraints, In *Proceedings of the European Control Conference*, pp. 2996–3001, 2009
10. P. A. Trodden, A. G. Richards, Robust distributed model predictive control using tubes. In *Proceedings of the American Control Conference*, pp. 2034–2039, 2006
11. P. A. Trodden, A. G. Richards. Robust distributed model predictive control with cooperation. In *Proceedings of the European Control Conference*, pp. 2172–2178, 2007
12. P. A. Trodden and A. G. Richards, Multi-vehicle cooperative search using distributed model predictive control, In *AIAA Guidance, Navigation, and Control Conference*, 2008
13. P.A. Trodden, A.G. Richards, Adaptive cooperation in robust distributed model predictive control, In *Proceedings of the IEEE Multi-conference on Systems and Control*, 2009
14. P.A. Trodden, A.G. Richards, Distributed model predictive control of linear systems with persistent disturbances. *International Journal of Control* **83**(8), 1653–1663 (2010)
15. A.N. Venkat, I.A. Hiskens, J.B. Rawlings, S.J. Wright, Distributed MPC strategies with application to power system automatic generation control. *IEEE Transaction on Control System Technology* **16**(6), 1192–1206 (2008)