

# Hierarchical cooperative distributed model predictive control

Brett T. Stewart, James B. Rawlings, and Stephen J. Wright

**Abstract**—Cooperative distributed model predictive control has recently been shown to provide stabilizing feedback for plants composed of any finite number of dynamically coupled subsystems [7]. No assumption is made about the strength of the coupling between the subsystems. This control scheme provides optimal feedback in the limit of infinite optimization iterates, can be terminated before convergence of the iterates, and guarantees stability and feasibility. These properties require communication between all subsystems in the plant at every iterate, however. In this paper, we weaken this requirement and propose an extension in which the subsystems are grouped in a hierarchy. The subsystems communicate with their neighbors at every iterate, but communicate with subsystems outside their neighborhood on a slower and asynchronous time schedule. We show that this extension is plantwide stabilizing.

We introduce a method to modify the information sent between neighborhoods in order to reduce the volume of communication and to hide input trajectories between neighborhoods. To achieve these properties, the leader of each subsystem performs a minor additional calculation at each plantwide information exchange. We conclude with an example demonstrating control performance in a chemical plant.

## I. INTRODUCTION

Chemical plants are composed of many interacting subsystems connected through a network of material, energy, and information streams. Interactions propagate through the network and all subsystems affect each other. One useful characterization of this network is the density of the inter-subsystem connections. Densely interconnected plants are complex and the interactions quickly spread throughout the network. Commonly, however, there exists some sparsity in the plant interconnections. In sparsely connected plants local interactions propagate quickly in groups of subsystems and affect the rest of the plant only on a longer time scale. For these plants, a network topology naturally arises in which subsystems are grouped into neighborhoods. Neighborhoods usually correspond to processing areas, in which several processes create raw materials for another set of subsystems. Performance in these plants is often improved using a hierarchy of controllers. Each neighborhood has a coordinating controller, which is itself coordinated with the other neighborhoods by a central plantwide controller [6]. Hierarchical control has the advantage of maintaining independence of the subsystems and neighborhoods, yet designing the hierarchical control scheme is often difficult. Model and objective mismatch occurs because the control is separated into layers.

We have shown previously [7] that cooperative control provides plantwide stability without a coordinator, but did not consider the underlying topology of the plant. The control scheme requires all subsystems to be synchronous and to communicate with all other subsystems at every

optimization iterate. These requirements limit the application of cooperative control to plants in which information can be reliably exchanged plantwide, and they do not allow integration of subsystems with multiple time scales. In this paper, we weaken these requirements by grouping the subsystems into hierarchies and allowing the subsystems to optimize and exchange information on *any* time scale. Our main contribution is to achieve the advantages of hierarchical control schemes without requiring additional coordinating controllers.

Previous work in the literature has considered groups of controllers from a geographical and physical perspective in which the control agents, usually autonomous vehicles, solve an optimization problem for both their own decision variables and their neighbor's decision variables [2]–[4]. These methods are a form of noncooperative control and do not provide stable feedback for plants with strong interactions between subsystems. The stability of cooperative control, alternatively, does not depend upon interaction strength [5, p.460]. Venkat provides an extension to cooperative control for time scale separation, in which subsystems are grouped by the sampling and computation time of the subsystems [8, pp.270-303]. This paper builds upon this concept, allowing any neighborhood grouping and communication frequency while guaranteeing exponential stability.

We begin in the next section with an introduction to the distributed optimization used for hierarchical model predictive control (MPC). In Section III, we provide definitions for hierarchical control, then show the control algorithm properties in Section IV. To simplify the exposition, we introduce these concepts with only four subsystems, but remark these results extend to any number of subsystems. In Section V we provide a physical definition of neighborhoods, which is utilized to reduce communication in the plant. We conclude with an example showing performance of a chemical plant.

### A. Notation

The symbol  $|\cdot|$  indicates the Euclidean 2-norm. The closed ball of radius  $r$  centered at the origin is denoted  $\mathbb{B}_r$ . The set  $\mathbb{I}_{l:m} = \{l, l+1, \dots, m-1, m\}$  for any  $l \leq m$ ; for  $l > m$ ,  $\mathbb{I}_{l:m} = \emptyset$ . We denote  $\phi(k, x)$  the closed-loop solution of the plant model at time  $k$  starting from state  $x$ .

## II. HIERARCHICAL OPTIMIZATION

Consider the optimization

$$\min_u V(u_1, u_2, u_3, u_4) \quad (1)$$

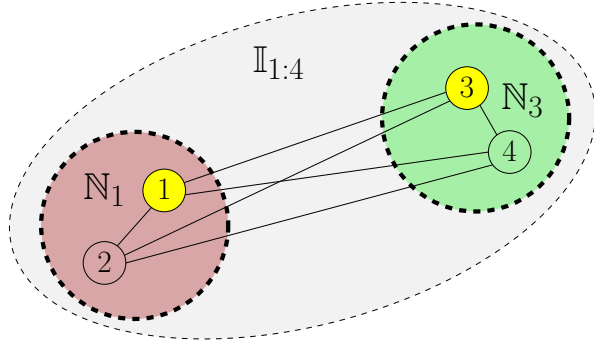


Fig. 1: Two level hierarchy: two neighborhoods, each with two subsystems such that  $\mathbb{N}_1 = \{1, 2\}$  and  $\mathbb{N}_3 = \{3, 4\}$ . The set of leaders  $\mathbb{L} = \{1, 3\}$ .

in which  $u_i \in \mathbb{R}$  and  $V : \mathbb{R}^4 \rightarrow \mathbb{R}_+$  is strictly convex. The optimization is performed as a Jacobi iteration [1, pp.219-223], in which there is a suboptimization performed for each variable and the solutions are traded between optimizers. We create a two-level hierarchy and define the neighborhoods  $\mathbb{N}_1 = \{1, 2\}$  and  $\mathbb{N}_3 = \{3, 4\}$  (see Fig. 1). At inner iterates, the suboptimizations utilize the latest iterates from variables within their own neighborhood, while the other variables are updated at outer iterates. Consider the optimization between outer iterates  $t$  and  $t + 1$ . Initializing  $v^0 \leftarrow u^t = (u_1^t, u_2^t, u_3^t, u_4^t)$ , at iterate  $p \in \mathbb{I}_{0:\bar{p}-1}$

$$\begin{aligned} v_1^* &= \arg \min_{v_1} V(v_1, v_2^p, u_3^t, u_4^t) & v_3^* &= \arg \min_{v_3} V(u_1^t, u_2^t, v_3, v_4^p) \\ v_2^* &= \arg \min_{v_2} V(v_1^p, v_2, u_3^t, u_4^t) & v_4^* &= \arg \min_{v_4} V(u_1^t, u_2^t, v_3^p, v_4) \end{aligned}$$

Inner iterates are generated by the convex step

$$v_i^{p+1} = w_i v_i^* + (1 - w_i) v_i^p \quad \forall i \in \mathbb{I}_{1:4}$$

in which for  $l \in \mathbb{L} = \{1, 3\}$ ,  $\sum_{j \in \mathbb{N}_l} w_j = 1$ ,  $w_i > 0 \forall i \in \mathbb{I}_{1:4}$ . After  $\bar{p}$  iterates, the outer iterate is performed

$$u_i^{t+1} = \lambda_i v_i^{\bar{p}} + (1 - \lambda_i) u_i^t \quad \forall i \in \mathbb{I}_{1:4}$$

in which the weighting  $\lambda_i$  is defined for each neighborhood  $\sum_{l \in \mathbb{L}} \lambda_l = 1$ ,  $\lambda_i = \lambda_j \forall i, j \in \mathbb{N}_l, \forall l \in \mathbb{L}$ .

*Lemma 1 (Convergence):* The cost  $V(u_1^t, u_2^t, u_3^t, u_4^t)$  is nonincreasing at each outer iterate  $t$  and converges as  $t \rightarrow \infty$ .

*Lemma 2 (Optimality):* The cost  $V(u_1^t, u_2^t, u_3^t, u_4^t)$  converges to the optimal value  $V(u_1^*, u_2^*, u_3^*, u_4^*)$ , and the iterates  $(u_1^t, u_2^t, u_3^t, u_4^t)$  converge to the optimizer  $(u_1^*, u_2^*, u_3^*, u_4^*)$  as  $t \rightarrow \infty$ .

### A. Example

To illustrate the cost decrease at each outer iterate, consider Fig. 2. In Fig. 2a, the optimizers in neighborhood 1 seek a solution to the optimization problem  $\min_{v_1, v_2} V(v_1, v_2, u_3^0, u_4^0)$  and after taking  $\bar{p} = 2$  steps, arrive at the suboptimal point  $\bar{v}_1^2 = (v_1^2, v_2^2)$ . Similarly in neighborhood 3, the suboptimal point  $\bar{v}_3^2 = (v_3^2, v_4^2)$  is found (see Fig. 2b). These two points are used to generate the next outer iterate  $u^1$ . We show this convex step in Fig. 2c, for which the x-axis and y-axis are, respectively, a representation of  $(u_1, u_2)$  and  $(u_3, u_4)$ . Notice that  $V(u^1) \leq V(u^0)$ .

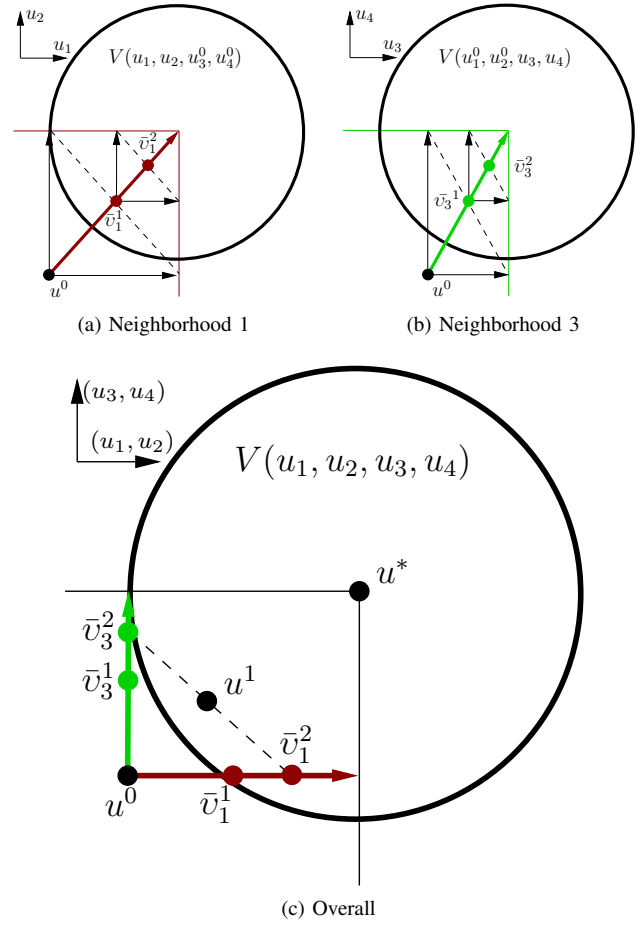


Fig. 2: Hierarchical optimization. Figs. (a) and (b) show each neighborhood's 2-dimensional optimization. Fig. (c) is a representation of the overall 4-dimensional optimization.

## III. PRELIMINARIES

We now provide definitions required to show properties of the control algorithm in the next section. For simplicity, we assume a plant composed of only four subsystems and relax this assumption in the sequel.

### A. Models

Consider the decentralized linear model

$$x_{ij}^+ = A_{ij} x_{ij} + B_{ij} u_j \quad \forall (i, j) \in \mathbb{I}_{1:4} \times \mathbb{I}_{1:4} \quad (2)$$

in which  $x_{ij} \in \mathbb{R}^{n_{ij}}$ ,  $u_j \in \mathbb{R}^{m_j}$ ,  $A_{ij} \in \mathbb{R}^{n_{ij} \times n_{ij}}$ , and  $B_{ij} \in \mathbb{R}^{n_{ij} \times m_j}$ . This model captures the effect from the inputs of subsystem  $j \in \mathbb{I}_{1:4}$  on the states of subsystem  $i \in \mathbb{I}_{1:4}$ . For notational simplicity, we collect the states to form the subsystem model for each  $i \in \mathbb{I}_{1:4}$

$$x_i^+ = A_i x_i + \sum_{j \in \mathbb{I}_{1:4}} \bar{B}_{ij} u_j$$

in which  $x_i = [x'_{i1} \dots x'_{i n_i}]' \in \mathbb{R}^{n_i}$ ,  $n_i = \sum_{j \in \mathbb{I}_{1:4}} n_{ij}$ ,  $A_i = \text{diag}(A_{i1}, \dots, A_{i n_i}) \in \mathbb{R}^{n_i \times n_i}$  and  $\bar{B}_{ij} = [0 \dots \underbrace{B'_{ij}}_{j \text{th position}} \dots 0]' \in \mathbb{R}^{n_i \times m_j}$ .

## B. Objective Functions

For each  $i \in \mathbb{I}_{1:4}$  define the quadratic stage cost and terminal penalty, respectively, as  $\ell_i(x_i, u_i) = \frac{1}{2}(x_i' Q_i x_i + u_i' R_i u_i)$  and  $V_{if}(x_i) = \frac{1}{2} x_i' P_{if} x_i$  in which  $Q_i \in \mathbb{R}^{n_i \times n_i}$ ,  $R_i \in \mathbb{R}^{m_i \times m_i}$ , and  $P_{if} \in \mathbb{R}^{n_i \times n_i}$ . For subsystem  $i \in \mathbb{I}_{1:4}$ , the objective function is

$$V_i(x_i(0), \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = \sum_{k=0}^{N-1} \ell_i(x_i(k), u_i(k)) + V_{if}(x_i(N))$$

in which  $\mathbf{u}_i = [u_i(0)', \dots, u_i(N-1)']' \in \mathbb{R}^{N m_i}$  and  $N > 0$  is the control horizon. We define the plantwide objective function

$$V(x(0), \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4) = \sum_{i \in \mathbb{I}_{1:4}} \rho_i V_i(x_i(0), \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$$

in which  $x = [x_1', \dots, x_4']'$  and  $\rho_i > 0$  for all  $i \in \mathbb{I}_{1:4}$ .

## C. Constraints

At each time step, the inputs satisfy the constraints  $u_i(k) \in \mathbb{U}_i, \forall k \in \mathbb{I}_{0:N-1}, \forall i \in \mathbb{I}_{1:4}$  in which  $\mathbb{U}_i$  is compact, convex, and contains the origin in its interior.

## D. Terminal Penalties and Terminal Constraints

Consider the real Schur decomposition

$$A_{ij} = \begin{bmatrix} S_{ij}^s & S_{ij}^u \end{bmatrix} \begin{bmatrix} A_{ij}^s & - \\ & A_{ij}^u \end{bmatrix} \begin{bmatrix} S_{ij}^{s'} \\ S_{ij}^{u'} \end{bmatrix}$$

in which  $A_{ij}^s$  and  $A_{ij}^u$  are composed of the stable and unstable blocks of  $A_{ij}$ , respectively. Defining  $S_i^s = \text{diag}(S_{i1}^s, \dots, S_{i4}^s)$  and  $A_i^s = \text{diag}(A_{i1}^s, \dots, A_{i4}^s)$  for each  $i \in \mathbb{I}_{1:4}$ , define

$$P_{if} = S_i^s \Sigma_i S_i^{s'} \quad (3)$$

in which  $\Sigma_i$  satisfies the Lyapunov equation  $A_i^{s'} \Sigma_i A_i^s - \Sigma_i = -S_i^{s'} Q_i S_i^s$ . We set the unstable modes of  $A_{ij}$  to zero at the end of the horizon with the constraint  $S_{ij}^{u'} x_{ij}(N) = 0$ .

## IV. CONTROL ALGORITHM

We now apply the optimization presented in Section II to distributed MPC. Consider the case of two neighborhoods, each with two subsystems, i.e.,  $\mathbb{N}_1 = \{1, 2\}$  and  $\mathbb{N}_3 = \{3, 4\}$  (see Fig. 1). With a slight abuse of notation we refer to  $\mathbb{N}_2 \leftarrow \mathbb{N}_1$  and  $\mathbb{N}_4 \leftarrow \mathbb{N}_3$ . We define a two-level hierarchy in which the subsystems in neighborhood  $\mathbb{N}_1$  communicate with each other at every time step, but for which they communicate with the subsystems in neighborhood  $\mathbb{N}_3$  only after  $N_s$  steps. For example, if all subsystems perform an outer iterate after every time step, then  $N_s = 0$ . As in Section II, we reoptimize this trajectory multiple times between outer iterates, but here only  $N_o$  input steps are optimized at a time.

### A. Initialization

For plant initialization, we require an initial condition  $\tilde{\mathbf{u}}_i$  for each subsystem  $i \in \mathbb{I}_{1:4}$

$$\tilde{\mathbf{u}}_i = \underbrace{[\tilde{u}_i(0)', \tilde{u}_i(1)', \dots, \tilde{u}_i(N_o - 1)']}_{N_o} \underbrace{[0, \dots, 0]'}_{N_s} \quad (4)$$

for which  $\tilde{\mathbf{u}}_i \in \mathbb{U}_i^N$  and the first  $N_o$  inputs satisfy the terminal constraint  $S_{ji}^{u'} x_{ji}(N_o) = 0$  for all  $j \in \mathbb{I}_{1:4}$ .

## B. Control Problem

Without loss of generality, let the outer iterate  $t = 0$ . We initialize  $\mathbf{v}_i^0 \leftarrow \mathbf{u}_i^0 \leftarrow \tilde{\mathbf{u}}_i$ . Between outer iterates, for  $k \in \mathbb{I}_{0:N_s}$ , subsystem  $i$  for every  $i \in \mathbb{I}_{1:4}$  solves the following optimization at inner iterate  $p \in \mathbb{I}_{k\bar{p}:(k+1)\bar{p}-1}$

$$\min_{\mathbf{v}_i} V(x(0), \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4) \quad (5a)$$

subject to

$$x_i^+ = A_i x_i + \sum_{j \in \mathbb{I}_{1:4}} \bar{B}_{ij} v_j \quad (5b)$$

$$\mathbf{v}_i \in \mathbb{U}_i^N \quad (5c)$$

$$S_{ji}^{u'} x_{ji}(k + N_o) = 0 \quad j \in \mathbb{I}_{1:4} \quad (5d)$$

$$|v_i| \leq d_i \sum_{j \in \mathbb{I}_{1:4}} |x_{ji}(0)| \quad \text{if } x_{ji}(0) \in \mathbb{B}_r \quad j \in \mathbb{I}_{1:4} \quad (5e)$$

$$v_j = v_j^p \quad j \in \mathbb{N}_i \setminus i \quad (5f)$$

$$v_l = v_l^0 \quad l \in \mathbb{I}_{1:4} \setminus \mathbb{N}_i \quad (5g)$$

$$v_i(\tau) = v_i^p(\tau) \quad \forall \tau \in \mathbb{I}_{0:k-1} \cup \mathbb{I}_{k+N_o:N-1} \quad (5h)$$

in which the radius  $r > 0$  of  $\mathbb{B}_r$  is chosen as small as desired.<sup>1</sup> Notice that although the horizon of inputs is  $N$ , only  $N_o$  of the inputs are decision variables at any time  $k$  due to constraint (5h). We denote the solution to these optimizations as  $\mathbf{v}_i^* = \mathbf{v}_i^*(x(0); \mathbf{v}_j^p; \mathbf{v}_l^0) \forall j \in \mathbb{N}_i \setminus i, \forall l \in \mathbb{I}_{1:4} \setminus \mathbb{N}_i$ . The next inner iterate is defined by

$$\mathbf{v}_i^{p+1} = w_i \mathbf{v}_i^* + (1 - w_i) \mathbf{v}_i^p \quad (6)$$

in which  $\sum_{j \in \mathbb{N}_i} w_j = 1$  and  $w_i > 0$ .

At time  $k \in \mathbb{I}_{0:N_s}$  each subsystem  $i \in \mathbb{I}_{1:4}$  takes  $\bar{p} > 0$  inner iterates and arrives at the point  $\mathbf{v}_i^{(k+1)\bar{p}}$ . Each subsystem  $i \in \mathbb{I}_{1:4}$  computes

$$\mathbf{u}_i^k = \lambda_i \mathbf{v}_i^{(k+1)\bar{p}} + (1 - \lambda_i) \tilde{\mathbf{u}}_i$$

in which  $\sum_{j \in \mathbb{L}} \lambda_j = 1, \lambda_i = \lambda_j \forall j \in \mathbb{N}_i$ , and  $\mathbb{L}$  is the set of all leaders, i.e.,  $\mathbb{L} = \{1, 3\}$ . Each subsystem injects the input  $u_i(k) = u_i^k(k)$ , the  $k$ th component of  $\mathbf{u}_i^k$ .

*Remark 3 (Weighting):* Whereas  $w_i$  measures the relative weight of a subsystem in  $\mathbb{N}_i$ ,  $\lambda_i$  measures the relative importance of neighborhood  $\mathbb{N}_i$  in the plantwide topology.

*Remark 4:* Between outer iterates, each subsystem input is reoptimized  $(N_s + 1)\bar{p}$  times.

*Lemma 5 (Feasibility):* The input trajectories satisfy  $(\mathbf{u}_1^k, \mathbf{u}_2^k, \mathbf{u}_3^k, \mathbf{u}_4^k) \in \mathbb{U}_1^N \times \mathbb{U}_2^N \times \mathbb{U}_3^N \times \mathbb{U}_4^N$  for all  $k \geq 0$ .

*Proposition 6 (Terminal constraint feasibility):* Given an input trajectory  $\mathbf{u}_i = [u_i(0)', \dots, u_i(k + N_o - 1)', 0, \dots, 0]'$  satisfying the constraint  $S_{ji}^{u'} x_{ji}(\tau + N_o) = 0 \forall j \in \mathbb{I}_{1:4}$  for  $\tau = k$ , the constraint is satisfied for all  $\tau \in \mathbb{I}_{k:N_s}$ .

The plant moves to the next time step, using  $\mathbf{v}_i^{(k+1)\bar{p}}$  as the initial condition for control problem (5) at time  $k + 1$ .

The point  $\mathbf{v}_i^{(k+1)\bar{p}}$  is feasible by Lemma 5 and Proposition 6.

Let  $\tilde{\mathbf{u}}_i^{N_s+} = (\tilde{\mathbf{u}}_1^{N_s+}, \tilde{\mathbf{u}}_2^{N_s+}, \tilde{\mathbf{u}}_3^{N_s+}, \tilde{\mathbf{u}}_4^{N_s+})$  denote the warm start, in which for  $i \in \mathbb{I}_{1:4}$

$$\tilde{\mathbf{u}}_i^{N_s+} = \underbrace{[u_i(N_s + 1)', \dots, u_i(N - 1)']}_{N_o - 1} \underbrace{[0, \dots, 0]'}_{N_s + 1} \quad (7)$$

<sup>1</sup>A detailed discussion of this constraint is provided in [7].

At  $k = N_s$ , the neighborhoods exchange  $\tilde{\mathbf{u}}_i^{N_s+}$  and initialize  $\mathbf{v}_i^{N_s+1} \leftarrow \tilde{\mathbf{u}}_i^{N_s+}$  for all  $i \in \mathbb{I}_{1:4}$ .

*Remark 7:* For  $N_s = 0$ , the controller performs like cooperative control [7] with only one optimization step taken between iterates.

### C. Objective Decrease

To prove stability, we show that the objective function decreases with every outer iterate and is bounded above between outer iterates.

*Lemma 8 (Upperbound):* Let  $t$  be the last outer iterate. Then the cost  $V(x, \mathbf{u}_1^k, \mathbf{u}_2^k, \mathbf{u}_3^k, \mathbf{u}_4^k)$  is upperbounded by  $V(x, \mathbf{u}_1^t, \mathbf{u}_2^t, \mathbf{u}_3^t, \mathbf{u}_4^t)$  for each  $k \in \mathbb{I}_{t:t+N_s}$ .

*Lemma 9 (Convergence and optimality):* The cost  $V(x, \mathbf{u}_1^t, \mathbf{u}_2^t, \mathbf{u}_3^t, \mathbf{u}_4^t)$  decreases at each outer iterate  $t$  and converges to the optimal value  $V^0(x)$  as  $t \rightarrow \infty$ . The inputs  $\mathbf{u}^t \rightarrow \mathbf{u}^0(x)$ , the optimal centralized input sequence, as  $t \rightarrow \infty$ .

*Remark 10 (Asynchronous):* Lemmas 8 and 9 do not require evenly spaced outer iterates, nor do they require  $\bar{p}$  be the same in each neighborhood.

### D. Stability

For every  $i \in \mathbb{I}_{1:4}$ , let  $\underline{\mathbf{A}}_i = \text{diag}(A_{1i}, \dots, A_{4i})$  and  $\underline{\mathbf{B}}_i = [B'_{1i}, \dots, B'_{4i}]'$

*Assumption 11:* For all  $(i, j) \in \mathbb{I}_{1:4} \times \mathbb{I}_{1:4}$

- 1) The systems  $(\underline{\mathbf{A}}_i, \underline{\mathbf{B}}_i)$  are stabilizable.
- 2) The input penalties  $R_i > 0$ .
- 3) The state penalties  $Q_i \geq 0$ .
- 4) The systems  $(A_i, Q_i)$  are detectable.
- 5)  $N \geq \max_{i \in \mathbb{I}_{1:4}}(\underline{n}_i^u)$ , in which  $\underline{n}_i^u$  is the number of unstable modes of  $\underline{\mathbf{A}}_i$ , i.e., number of  $\lambda \in \text{eig}(\underline{\mathbf{A}}_i)$  such that  $|\lambda| \geq 1$ .

Notice we make no assumption on the strength of the interactions between subsystems. Let  $\mathbb{X}_N$  be the forward invariant set of all initial states such that the control problem (5) is feasible.

*Theorem 12 (Exponential stability):* Given Assumption 11, the origin of the closed-loop system  $\phi(k, x)$  is exponentially stable on the set  $\mathbb{X}_N$ .

*Proof:* The proof closely follows that of the discussion of suboptimal MPC in [5, pp.415-420]. Consider the initialization of the control algorithm at the next outer iterate with the warm start

$$\begin{aligned} V(x^{N_s+}, \tilde{\mathbf{u}}^{N_s+}) &= V(x, \mathbf{u}^{N_s}) \\ &- \sum_{i \in \mathbb{I}_{1:4}} \sum_{k=0}^{N_s} \frac{\rho_i}{2} \ell_i(x_i(k), u_i(k)) \\ &+ \sum_{i \in \mathbb{I}_{1:4}} \frac{\rho_i}{2} x_i(N)' \Psi_i(N_s) x_i(N) \end{aligned} \quad (8)$$

in which  $x^{N_s+} = \phi(N_s + 1; x)$  and

$$\Psi_i(N_s) = (A_i^{N_s+1})' P_{if} A_i^{N_s+1} - P_{if} + \sum_{k=0}^{N_s} (A_i^k)' Q_i A_i^k$$

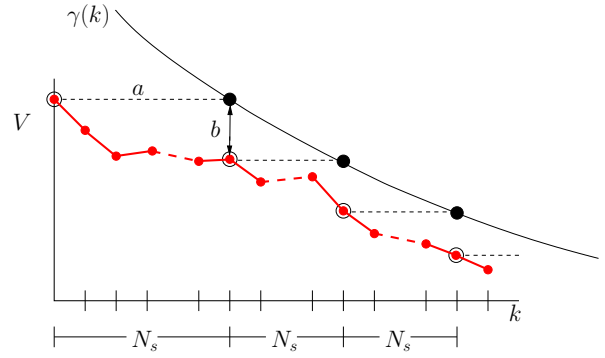


Fig. 3: Exponential decaying upperbound of objective function. The upperbound given by line (a) is shown by Lemma 8. The cost drop given by the line (b) is show by Lemma 9. These properties allow the construction of the exponentially decaying function  $\gamma(k)$ .

By the terminal constraint (5d) and the terminal penalty (3),  $x_i(N)' \Psi(N_s) x(N) = 0, \forall i \in \mathbb{I}_{1:4}$  therefore

$$V(x^{N_s+}, \tilde{\mathbf{u}}^{N_s+}) = V(x, \mathbf{u}^{N_s}) - \sum_{i \in \mathbb{I}_{1:4}} \sum_{k=0}^{N_s} \frac{\rho_i}{2} \ell_i(x_i(k), u_i(k))$$

By Lemma 9

$$\begin{aligned} V(x^{N_s+}, \tilde{\mathbf{u}}^{N_s+}) &\leq V(x, \mathbf{u}^{N_s}) - \sum_{i \in \mathbb{I}_{1:4}} \sum_{k=0}^{N_s} \frac{\rho_i}{2} \ell_i(x_i(k), u_i(k)) \\ &\leq V(x, \mathbf{u}^{N_s}) - \sum_{i \in \mathbb{I}_{1:4}} \frac{\rho_i}{2} \ell(x_i, u_i(0)) \end{aligned}$$

Hence, there exists a  $c$  such that

$$V(x^{N_s+}, \tilde{\mathbf{u}}^{N_s+}) - V(x, \mathbf{u}^{N_s}) \leq -c |x, u(0)|^2 \quad (9)$$

Between outer iterates, by Lemma 8

$$V(x, \mathbf{u}^k) - V(x, \mathbf{u}^0) \leq 0 \quad k \in \mathbb{I}_{0:N_s}$$

The standard upper and lower bounding norms on  $V(\cdot)$  follow from the compactness of each  $\mathbb{U}_i$ . We remove the appearance of the input  $u(0)$  in the norm of (9) using constraint (5e) and complete the proof similarly to Lemma 6.4 in [5, p.418]. A visual representation of the proof is shown in Fig. 3. Therefore we may construct an exponentially decaying function upperbounding  $V$  at each time step, and exponential stability follows. ■

*Remark 13 (M Subsystems and L Levels):* The above arguments are presented with a plant composed of 4 subsystems split into 2 neighborhoods. These arguments are extended to any number of subsystems, however, by replacing 4 in the above arguments with any integer  $M > 0$ . Levels are added to the hierarchy by treating the outer iterate for level  $L$  as an inner iterate for level  $L + 1$ .

## V. REDUCING COMMUNICATION

In addition to reducing the frequency of information exchange between subsystems, it is also advantageous to reduce the complexity of the plantwide communication topology. In this section, we explicitly define neighborhoods and provide a change of variables that allows exchange of information between only a subset of the subsystems.

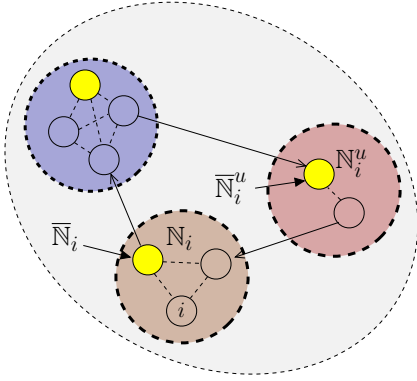


Fig. 4: Subsystems grouped into neighborhoods with leaders. The neighborhood for subsystem  $i$  is  $\mathbb{N}_i$  with leader  $\bar{\mathbb{N}}_i$ . The neighborhood upstream of subsystem  $i$  is  $\mathbb{N}_i^u$  with leader  $\bar{\mathbb{N}}_i^u$ .

### A. Time Delays

Consider the model

$$\begin{bmatrix} x_{ij} \\ z_{ij} \end{bmatrix}^+ = \begin{bmatrix} A_{ij} & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{ij} \\ z_{ij} \end{bmatrix} + \begin{bmatrix} 0 \\ B_{ij} \end{bmatrix} u_j$$

In this model, one time step elapses before a change in  $u_j$  contributes to  $x_{ij}$ . We then say that the  $x_{ij}$  model contains one *time delay* and abbreviate the model as

$$x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j^-$$

Similarly, a model with  $q$  time delays is abbreviated  $x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j^{q-}$ .

### B. Network Structure

**Definition 14 (Neighborhood):** The set  $\mathbb{N}$  is denoted a *neighborhood* if for all  $(i, j) \in \mathbb{N} \times \mathbb{N}$  the  $x_{ij}$  model has *zero* time delays.

**Definition 15 (Upstream Neighborhood):** The neighborhood  $\mathbb{N}^u$  is defined as *upstream* to neighborhood  $\mathbb{N}$  if, for all  $i \in \mathbb{N}$  and  $j \in \mathbb{N}^u$ , the model  $x_{ij}$  has *one* time delay.

For all  $i \in \mathbb{I}_{1:M}$ , denote the neighbors of subsystem  $i$  as  $\mathbb{N}_i$  and the upstream neighbors as  $\mathbb{N}_i^u$ . For each neighborhood  $\mathbb{N}_i$ , we denote the (singlet) leader set for the neighborhood as  $\bar{\mathbb{N}}_i$  for all  $i \in \mathbb{I}_{1:M}$ . Similarly, the leader set for the upstream neighborhoods of subsystem  $i$  is denoted  $\bar{\mathbb{N}}_i^u$ . For example, see Fig. 4.

### C. Models

Consider an alternative model to (2) in which for every  $i \in \mathbb{I}_{1:M}$ , the model for subsystem  $i$  is

$$\begin{aligned} x_{ij}^+ &= A_{ij}x_{ij} + B_{ij}u_j & \forall j \in \mathbb{N}_i \\ x_{il}^+ &= A_{il}x_{il} + B_{il}u_l^- & \forall l \in \mathbb{N}_i^u \\ x_{is}^+ &= A_{is}x_{is} + B_{is}u_s^{q-} & \forall s \in \mathbb{N}_i^{[q]}, \forall q > 1 \end{aligned}$$

in which the  $q$ th upstream neighborhood is  $\mathbb{N}_i^{[q]} = \bigcup_{j \in \mathbb{N}_i^u} \mathbb{N}_j^{[q-1]}$  with  $\mathbb{N}_i^{[1]} = \mathbb{N}_i^u$ . Without loss of generality, we may write the model in the following form

$$x_{ij}^+ = A_{ij}x_{ij} + B_{ij}u_j \quad \forall j \in \mathbb{N}_i \quad (10a)$$

$$x_{il}^+ = A_{il}x_{il} + E_{il}x_{sl} \quad \forall l \in \mathbb{N}_i^u, s \in \bar{\mathbb{N}}_l \quad (10b)$$

in which  $E_{il} \in \mathbb{R}^{n_{ij} \times n_{sl}}$ . Collecting the states into vectors  $x_i = [\dots, x_{ij}, \dots]'$  for all  $i \in \mathbb{I}_{1:M}, j \in \mathbb{N}_i \cup \mathbb{N}_i^u$ , and we form the following matrices for all  $i \in \mathbb{I}_{1:M}$

$$\begin{aligned} \bar{A}_{ii} &= \text{diag}(\dots, A_{ij}, \dots) \quad \forall j \in \mathbb{N}_i \cup \mathbb{N}_i^u \\ \bar{A}_{il} &= [\dots, \bar{E}'_{ij}, \dots, 0, \dots, 0]' \quad \forall j \in \mathbb{N}_l \text{ s.t. } l \in \bar{\mathbb{N}}_i^u \end{aligned}$$

in which  $\bar{A}_{ij} \in \mathbb{R}^{n_i \times n_j}$  and

$$\bar{E}'_{ij} = [0 \ \dots \ \underbrace{E'_{ij}}_{j\text{th position}} \ \dots \ 0]' \in \mathbb{R}^{n_i \times n_{ij}} \quad \forall j \in \mathbb{N}_i^u$$

For all  $\bar{A}_{ij}$  not defined above,  $\bar{A}_{ij} = 0$ . These matrices form the model

$$x_i^+ = \bar{A}_{ii}x_i + \sum_{j \in \mathbb{N}_i} \bar{B}_{ij}u_j + \sum_{l \in \bar{\mathbb{N}}_i^u} \bar{A}_{il}x_l \quad \forall i \in \mathbb{I}_{1:M} \quad (11)$$

Notice that each state is affected only by inputs within its neighborhood and the states of upstream leaders.

### D. Interaction Graph

To represent the state-to-state interactions of (11), we construct a directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  in which each subsystem is a vertex, i.e.,  $\mathcal{V} = \mathbb{I}_{1:M}$ . For each subsystem  $i \in \mathbb{I}_{1:M}$ , there exists an edge  $(i, i) \in \mathcal{E}$  and an edge  $(j, i) \in \mathcal{E}$  for all  $j \in \bar{\mathbb{N}}_i^u$ .

A path is a sequence of vertices connected by edges. A path  $P = \{i, j, l, \dots, m, n\}$  such that  $i, j, l, \dots, m, n \in \mathcal{V}$  and  $(i, j), (j, l), \dots, (m, n) \in \mathcal{E}$ . Let the path  $P_{ij}$  be a path from subsystem  $i$  to subsystem  $j$  and the set  $\mathbb{P}_{ij}^k = \{P_{ij}\}$  be the set of all paths from  $i$  to  $j$  having  $k$  connecting edges. Define

$$\bar{A}_{ij}^{[k]} = \sum_{P_{ij} \in \mathbb{P}_{ij}^k} \prod_{(m,n) \in P_{ij}} \bar{A}_{mn}$$

This matrix gives the total effect of the state-to-state interactions from subsystem  $i$  to subsystem  $j$  that propagate in  $k$  steps.

### E. Reduced Communication

Communication is reduced by rearranging model (11) in the following way. For all  $i \in \mathbb{I}$

$$\alpha_i^+ = \bar{A}_{ii}\alpha_i + \sum_{j \in \mathbb{N}_i} \bar{B}_{ij}u_j \quad (12)$$

for which  $\alpha_i(0) = x_i(0)$ . Notice that  $\alpha$  is defined only for the neighborhood leaders, and its computation needs only information, e.g., states and inputs, available within the neighborhood. At time  $k \geq 0$ , the states of subsystem  $i \in \mathbb{I}_{1:M}$  are

$$\begin{aligned} x_i(k) &= \bar{A}_{ii}^k x_i(0) + \sum_{\tau=0}^{k-1} \sum_{j \in \mathbb{N}_i} \bar{A}_{ii}^{k-\tau-1} \bar{B}_{ij} u_j(\tau) \\ &+ \sum_{\tau=0}^{k-1} \sum_{l \in \mathbb{L}} \sum_{s \in \mathbb{I}_{1:M} \setminus l} \bar{A}_{is}^{[k-\tau-1]} \bar{A}_{sl} \alpha_l(\tau) \end{aligned} \quad (13)$$

To compute its state, each subsystem communicates only with the neighborhood leaders. This communication topology reduces the complexity of information sharing in the network.

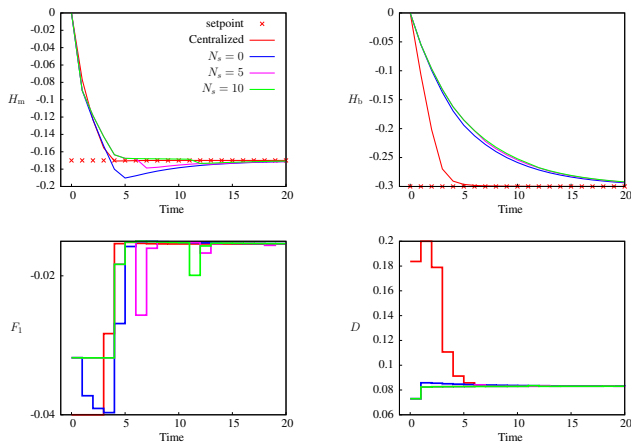


Fig. 5: Performance of hierarchical MPC in a two reactor and column plant for various communication frequencies.

### F. Control Algorithm

As previously, assume the last outer iterate is  $k = 0$ . First the leaders communicate and compute the values of  $\alpha_l^0$  for all  $l \in \mathbb{L}$ . Then for all  $i \in \mathbb{I}_{1:M}$ , the following optimization is performed for all  $p \in \mathbb{I}_{k\bar{p}:(k+1)\bar{p}}$

$$\min_{\mathbf{v}_i} V(x(0), \mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_M) \quad (14a)$$

subject to (13)

$$\mathbf{v}_i \in \mathbb{U}_i^N \quad (14b)$$

$$S_{ji}^{u,l} x_{ji}(k + N_o) = 0 \quad j \in \mathbb{I}_{1:M} \quad (14c)$$

$$|\mathbf{v}_i| \leq d_i \sum_{j \in \mathbb{I}_{1:M}} |x_{ji}(0)| \quad \text{if } x_{ji}(0) \in \mathbb{B}_r \quad j \in \mathbb{I}_{1:M} \quad (14d)$$

$$\mathbf{v}_j = \mathbf{v}_j^p \quad j \in \mathbb{N}_i \setminus i \quad (14e)$$

$$\mathbf{v}_l = \mathbf{v}_l^0 \quad l \in \mathbb{I}_{1:M} \setminus \mathbb{N}_i \quad (14f)$$

$$v_i(\tau) = v_i^p(\tau) \quad \forall \tau \in \mathbb{I}_{0:k-1} \cup \mathbb{I}_{k+N_o:N-1} \quad (14g)$$

The inner and outer iterates are performed as in Section IV. At an outer iterate, the leaders recompute  $\alpha$ . The procedure is then repeated.

*Remark 16:* Because the neighborhoods exchange the trajectory of  $\alpha$  values instead of the input trajectories, the future input plans are not shared between neighborhoods.

*Remark 17:* The control algorithm and optimization problem remains the same and, given the previous assumptions remain satisfied, all properties proven in the previous sections are maintained.

## VI. EXAMPLE

We provide as an example the setpoint change of a two reactor and distillation column system proposed by Venkat [9]. Here we group the two reactors into a neighborhood, and reduce the times at which they trade input trajectories with the distillation column. We show the performance in Fig. 5 and give the objective function values in Table I. Although performance degrades as the subsystems wait longer to communicate, i.e., by increasing  $N_s$ , closed-loop stability is maintained, and the performance loss is not significant.

	Cost
Centralized	0.95
Cooperative (1 iterate): $N_s = 0$	1.60
$N_s = 1$	1.633
$N_s = 5$	1.661
$N_s = 10$	1.669
$N_s = 50$	1.670

TABLE I: Hierarchical MPC performance

## VII. CONCLUSION

In this paper, we expand cooperative distributed model predictive control to use hierarchies, but without requiring new levels of coordinating controllers. This extension reduces the frequency of communication between subsystems. We develop a method to limit the communication further by a changing variables and defining a hierarchy of neighborhoods and leaders. Both of these strategies inherit the properties of cooperative control, namely, feasibility of input constraints and exponential stability for any number of optimization iterates. We conclude with an example showing the performance degradation associated with increasing the time between information exchanges.

## VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of the industrial members of the Texas-Wisconsin-California Control Consortium. The authors would like to thank Prof. Bart De Schutter and Prof. Tamas Keviczky for their helpful discussions.

## REFERENCES

- [1] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation*. Athena Scientific, Belmont, Massachusetts, 1997.
- [2] William B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Auto. Cont.*, 52(7):1249–1263, 2007.
- [3] William B. Dunbar and Richard M. Murray. Distributed receding horizon control with application to multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.
- [4] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G.J. Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Trans. Cont. Sys. Tech.*, 16(1):19–33, Jan. 2008.
- [5] James B. Rawlings and David Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI, 2009. 576 pages, ISBN 978-0-9759377-0-9.
- [6] Riccardo Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *J. Proc. Cont.*, 19(5):723–731, May 2009.
- [7] Brett T. Stewart, Aswin N. Venkat, James B. Rawlings, Stephen J. Wright, and Gabriele Pannocchia. Cooperative distributed model predictive control. *Sys. Cont. Let.*, 59:460–469, 2010.
- [8] Aswin N. Venkat. *Distributed Model Predictive Control: Theory and Applications*. PhD thesis, University of Wisconsin–Madison, October 2006.
- [9] Aswin N. Venkat, James B. Rawlings, and Stephen J. Wright. Stability and optimality of distributed, linear MPC. Part 1: state feedback. Technical Report 2006–03, TWGCC, Department of Chemical and Biological Engineering, University of Wisconsin–Madison (Available at <http://jbrwww.che.wisc.edu/tech-reports.html>), October 2006.