Limited-Communication Distributed Model Predictive Control for Coupled and Constrained Subsystems

Rawand E. Jalal and Bryan P. Rasmussen

Abstract—This brief presents the limited-communication distributed model predictive control (LC-DMPC) algorithm for coupled and constrained linear discrete systems. In this framework, the global control problem is divided into a number of coupled subsystems based on a neighbor upstream and downstream structure. According to this structure, a subsystem views the coupling signals from upstream neighbors as measured disturbances and at the same time has outputs to downstream neighbors. In contrast with most DMPC schemes, the individual subsystems solve a different cost than the centralized problem. At each iteration, two bidirectional signals are communicated: predicted disturbances for downstream neighbors and local cost sensitivity to disturbances from upstream neighbors. With only neighbor-to-neighbor communication, the LC-DMPC scheme can converge to the centralized optimum without sharing dynamics or costs between the distributed agents. The closed-loop stability is guaranteed by assuming sufficiently long horizons. To reduce the computational burden, local control actions are parameterized by Laguerre functions. This leads to smaller distributed control problems and allows more iterations per sampling. A coupled tank process demonstrates the main aspects of the proposed algorithm.

Index Terms—Coupled tank process, distributed model predictive control (DMPC), Laguerre functions, model-based predictive control.

I. INTRODUCTION

Effective control for large-scale systems requires coordination to guarantee operating at optimal conditions within constraints and ensure safety. This motivates researchers to develop reliable control algorithms that feature certain levels of modularity and can handle real industrial limitations, such as distributed components, constraints, and multivariable processes. The general structure of model predictive control (MPC) can deal with most of these limitations explicitly, which makes it a popular control strategy for numerous industrial processes. In many cases, a fully centralized MPC has been implemented successfully. However, for plants with large numbers of variables or widely distributed subsystems, the computational time and communication loads may impede the controller in making real-time decisions. In contrast, completely decentralized MPCs might not be the workaround solution either. In this framework, an independent MPC is designed by ignoring couplings between subsystems. For weak interacting processes, decentralized MPCs may be able to overcome the interactions if they are tuned deliberately. For strong couplings, however, the interactions can lead to loss in performance and may not even stabilize the network [1]. Alternatively, the control problem of network processes can be approached by distributed MPC (DMPC). In this context, the large-scale plant is decomposed into a number of subsystems that consider couplings and each subsystem has its own controller. Seeking the centralized performance, the controllers cooperate or share information. Various DMPC algorithms have been proposed in the literature and widely referenced overviews can be found in [2] and [3].

In DMPC algorithms, the key point is the condition that guarantees convergence to systemwide solutions and local closed-loop stability. To realize this condition, most of the DMPC approaches in the literature require either a full communication setup, a solution of a centralized problem, or sharing of dynamic information with local agents. In a full information sharing setup, the algorithm needs to share data with all agents in the network. Venkat et al. [4] introduced a fully connected DMPC approach that requires complete information about the network with the intention of satisfying the global control objective. In order to have stabilizing local terminal costs and sets, Venkat [5] recommends solving a centralized linear quadratic regulator problem. This problem needs to be recomputed whenever a setpoint changes or a model is updated. As an alternative method, Conte et al. [6] proposed a distributed synthesis of local terminal costs and sets for coupled discrete linear systems. In this synthesis, local agents need to know the dynamics of the couplings with neighbors. Cai et al. [7] developed a singular value decomposition-based DMPC framework that converges to a global solution for a fully connected structure. For stochastic linear discrete systems, Barcelli et al. [8] introduced an LMI solution as centralized synthesis of the distributed regulators and for nonlinear continuous processes, Liu et al. [9] proposed a fully connected Lyapunov-based DMPC approach.

To reduce the communication burden, neighbor-to-neighbor communication frameworks have been suggested. In this structure, only coupled agents are required to share information. Farina and Scattolini [10] introduced a neighbor-to-neighbor communication-based DMPC algorithm for linear discrete systems. The closed-loop stability and the convergence were proven by solving a centralized stabilizing matrix. Dunbar [11] also used the neighbor-to-neighbor sharing scheme for continuous nonlinear systems where, for a feasible and stable local controller, a system-wide problem has to be solved. The same
information exchange structure is used by Liu and Liu [12] for the iterative and noniterative DMPC structures to control nonlinear coupled continuous systems. Once again, the local closed-loop stability is proven based on a condition on the systemwide initial conditions. Müller et al. [13] proposed a DMPC framework for decoupled discrete nonlinear systems that are coupled in constraints or objective indices. Only coupled agents are required to cooperate. For recursive feasibility and stability, the overall closed-loop system state has to be inside a predefined terminal systemwide region.

In this brief, a novel limited-communication DMPC (LC-DMPC) scheme based on neighbor-to-neighbor information sharing is introduced. It is an iterative and cooperative algorithm for linear discrete systems where convergence requires only coupled subsystems with local information to cooperate without a solution of any centralized problem. During an iteration, a local controller shares its predicted effects with local neighbors and receives the sensitivity that the neighbors’ cost functions have for these effects at the next iteration. Then, the controller solves an updated problem by minimizing the summations of a deviation effects at the next iteration. Then, the controller solves an updated problem by minimizing the summations of a deviation effects with local neighbors and receives the predicted data (4) between the local controllers. Then, the relation between the network inputs and outputs can be written as

\[ V_i = [v_i^T(k) \, v_i^T(k+1) \, \ldots \, v_i^T(k+N_p,i-1)]^T. \]

Similarly, define

\[ Z_i = [z_i^T(k) \, z_i^T(k+1) \, \ldots \, z_i^T(k+N_p,i-1)]^T. \]

Assume, for simplicity, all subsystems in the network have the same prediction horizons equal to \( N_p \); then, the network disturbance inputs and outputs can be formulated as

\[ V_{r_i} = [V_1^T \, V_2^T \, \ldots \, V_p^T]^T, \]

\[ Z_{p_{r,i}} = [Z_1^T \, Z_2^T \, \ldots \, Z_p^T]^T, \]

with \( r_i = N_p \cdot \sum_{j=1}^{p} p_{c,i} \) and \( p_{c} = N_p \cdot \sum_{i=1}^{p} p_{c,i} \).

For theoretical analysis only, an interconnecting matrix \( \Gamma \in [0,1] \) with a dimension of \( r_i \times p_c \) is defined to manage the communication of the predicted data (4) between the local controllers. Then, the relation between the network inputs and outputs can be written as

\[ V = \Gamma Z. \]

The structure of the interconnecting matrix must reveal the actual coupling between the subsystems. That is, for each disturbance output \( Z_i \), there is a corresponding input vector \( V_i \) (i.e., \( r_i = p_c \)). As an example, in Fig. 2(a), a six-tank process is given where the tanks are coupled in states and control actions. By defining three subsystems as shown in Fig. 2(b) and following the defined variables in Section V, the couplings between the subsystems are defined as: subsystem \( \Sigma_2 \) affects both subsystems \( \Sigma_1 \) and \( \Sigma_3 \) through \( u_2 \) and \( x_5 \) and has four disturbances coming from the same subsystems given by \( u_1, x_4, u_3 \), and \( x_6 \), respectively. Therefore, for \( N_p \) equal to 1

\[ Z_2 = [Z_{2,1}^T \, Z_{2,3}^T]^T = [(x_5 \, u_2) \, (x_5 \, u_2)]^T, \]

\[ V_2 = [V_{2,1}^T \, V_{2,3}^T]^T = [(x_4 \, u_1) \, (x_6 \, u_3)]^T. \]
and the network disturbance inputs

Thus

The couplings between the defined subsystems are shown in Fig. 2(b). Then, the network disturbance outputs would be

and the network disturbance inputs V is related to Z through the interconnecting matrix \( \Gamma \) as follows:

This six-tank process can also be coupled through the control actions only and matrix \( \Gamma \) can be manipulated to handle such coupling. This concept is easily extended for \( n \) numbers of coupled tanks with any value of \( N_p \). The example also shows that any two-coupled plants can simultaneously be upstream and downstream neighbors, where the neighbors are defined based on the flow of the coupled signals.

**B. Local Subsystem Predicted Models**

By the repeated application of (1) along \( N_p \), the predicted future dynamics for vectors \( Y_i \) and \( Z_i \) can be written as

\[
Y_i = F_{y,i}X_{0,i}(k) + M_{y,i}U_i + N_{y,i}V_i
\]

\[
Z_i = F_{z,i}X_{0,i}(k) + M_{z,i}U_i + N_{z,i}V_i
\]

where

\[
F_{y,i} = \left[ (C_{y,i}A_i)^T (C_{y,i}A_i^2)^T \cdots (C_{y,i}A_i^{N_p})^T \right]^T
\]

\[
F_{z,i} = \left[ C_{z,i}(C_{z,i}A_i) \cdots (C_{z,i}A_i^{N_p-1})^T \right]^T
\]

\[
M_{y,i} = \begin{bmatrix}
C_{y,i}B_{u,i} & 0 & \cdots & 0 \\
C_{y,i}A_iB_{u,i} & C_{y,i}B_{u,i} & 0 & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
C_{y,i}A_i^{N_p-1}B_{u,i} & C_{y,i}A_i^{N_p-2}B_{u,i} & \cdots & C_{y,i}B_{u,i} \\
\end{bmatrix}
\]

\[
M_{z,i} = \begin{bmatrix}
D_{z,i} \\
C_{z,i}B_{u,i} \\
\vdots \\
C_{z,i}A_i^{N_p-2}B_{u,i} & C_{z,i}A_i^{N_p-3}B_{u,i} & \cdots & C_{z,i}B_{u,i} \\
\end{bmatrix}
\]

and \( N_{y,i} \) and \( N_{z,i} \) are similar to \( M_{y,i} \) and \( M_{z,i} \) with \( B_{u,i} \) replacing \( B_{u,i} \) for \( N_{y,i} \) and with \( D_{z,i} = 0 \) for \( N_{z,i} \), respectively. Also

\[
U_i = [u_i(k) \ u_i(k+1) \ \cdots \ u_i(k+N_p-1)]^T.
\]

**C. Centralized and Distributed Cost Functions**

In this brief, the centralized controller is solving a typical MPC problem, namely, a linearly constrained quadratic cost function that penalizes the sum of deviation from a desired reference and control inputs along \( N_p \). On the other hand, a local controller is minimizing the sum of deviation from a desired local reference, the local control efforts, and its effects on downstream neighbors. That is, agent \( i \) is solving a cost function with penalties on its effects for downstream subsystems. Let \( q_i \in \mathbb{R}^{P_{y,i} \times p_y,i} > 0 \) and \( s_i \in \mathbb{R}^{N_{u,i} \times N_{r,i}} > 0 \) be two constant matrices that serve as local weights on the predicted errors and control actions, respectively, and define the local error signal along \( N_p \) as \( e_i = (r_i(k) - Y_i) \) with

\[
r_i(k) = [r_i(k+1) \ r_i(k+2) \ \cdots \ r_i(k+N_p)]^T
\]

where \( r_i \) is a desired local reference that must be reachable by the local controller with local constraints. Reachable references can be computed in the centralized level by a trajectory planner [14] and the result is then distributed.

To state the centralized problem, for a network of \( p \) coupled subsystems, the following vectors: \( Y = [Y_1^T \ Y_2^T \ \cdots \ Y_p^T]^T \) and similarly for the reference \( r(k) \), error signal \( e \), control action \( U \), and initial condition \( X_0(k) \), and block-diagonal matrices: \( Q = \text{diag}(Q_1, Q_2, \ldots, Q_p) \), \( S = \text{diag}(S_1, S_2, \ldots, S_p) \), \( F_Y = \text{diag}(F_{Y,1}, F_{Y,2}, \ldots, F_{Y,p}) \), where \( Q_i \) and \( S_i \) are given by

\[
Q_i = \text{diag}(q_i(k+1), q_i(k+2), \ldots, q_i(k+N_p))
\]

\[
S_i = \text{diag}(s_i(k+1), s_i(k+2), \ldots, s_i(k+N_p))
\]

and \( F_Y, M_Y, M_Z, N_Y, \) and \( N_Z \) are given in the same way as \( F_Y \), are defined along \( N_p \). Thus, the network regulated output \( Y \) and disturbance output \( Z \) can be formulated as

\[
Y = F_YX_0(k) + M_YU + N_YV
\]

\[
Z = F_ZX_0(k) + M_ZU + N_ZV.
\]
Through (9), the network disturbance input $V$ can be written as

$$V = FZ = (I - \Gamma N_z)\Gamma (F_z X_0(k) + M_z U) W = W(F_z X_0(k) + M_z U).$$

With constraints imposed only on the control actions, the systemwide optimization problem is defined to be

$$\begin{align*}
\min J &= e^T Q e + U^T S U \\
\text{s.t.} \quad Y &= F_y X_0(k) + M_y U + N_y W(F_z X_0(k) + M_z U) \\
U_{\min} &\leq U \leq U_{\max}.
\end{align*}$$

This problem can also be represented as

$$\begin{align*}
\min J &= U^T H U + 2U^T F \\
\text{s.t.} \quad A_i U &\leq B_i
\end{align*}$$

where

$$\begin{align*}
H &= S + (M_y + N_y W M_z)^T Q (M_y + N_y W M_z) \\
F &= (M_y + N_y W M_z)^T Q ((F_y + N_y F_z X_0(k)) - (M_y + N_y W M_z)^T Q r(k) \\
A_i &= \text{diag}(I_{r_u,N_p}, \ldots, I_{r_u,N_p}) \\
B_i &= \left[ \begin{array}{c}
U_i^{T_{\max}} \\
U_i^{T_{\min}} \\
\vdots \\
U_i^{T_{\max}} \\
U_i^{T_{\min}}
\end{array} \right]^T.
\end{align*}$$

For the distributed optimization problem, each distributed controller is assigned to solve the following local problem:

$$\begin{align*}
\min J_i &= e_i^T Q_i e_i + U_i^T S_i U_i + \Psi_i^T Z_i \\
\text{s.t.} \quad \text{Local dynamics (6) and (7)} \\
U_{\min} &\leq U_i \leq U_{\max}.
\end{align*}$$

The vector $\Psi_i$ represents the sensitivity of downstream neighbors to the disturbance coming from the $i$th plant. Once again this problem can be reformulated as

$$\begin{align*}
\min J_i &= U_i^T H_i U_i + 2U_i^T F_i + V_i^T E_i V_i + 2V_i^T T_i \\
\text{s.t.} \quad A_{i_{uq}} U_i &\leq B_{i_{uq}}
\end{align*}$$

where

$$\begin{align*}
H_i &= M_{x,i} Q_i M_{y,i} + S_i, \quad E_i = N_{y,i} Q_i N_{y,i} \\
F_i &= M_{x,i} Q_i [F_y x_0(i,k) + N_y V_i - t_i(k)] + 0.5 M_{x,i}^T \Psi_i \\
T_i &= N_{y,i}^T Q_i [F_y x_0(i,k) - t_i(k)] + 0.5 N_{x,i}^T \Psi_i \\
A_{i_{uq}} &= \text{diag}(I_{r_u,N_p}, \ldots, I_{r_u,N_p}) \\
B_{i_{uq}} &= \left[ \begin{array}{c}
U_i^{T_{\max}} \\
U_i^{T_{\min}}
\end{array} \right]^T.
\end{align*}$$

The value of $\Psi_i$ is computed and communicated by downstream agents. More specifically, denote the current subsystem as $\Sigma_i$ and downstream subsystem as $\Sigma_{i+1}$. If $\Sigma_{i+1}$ receives the disturbance $V_{i+1}$ from $\Sigma_i$ ($V_{i+1} = Z_i$) (see Fig. 3), it computes the sensitivity of its cost with respect to $V_{i+1}$ as

$$\gamma_{i+1} = \frac{\partial J_{i+1}}{\partial V_{i+1}} = 2E_i V_{i+1} + T_{i+1} + N_{y,i+1} Q_{i+1} M_{y,i+1} U_{i+1}$$

and shares the values of $\gamma_{i+1}$ with $\Sigma_i$, i.e., $\Psi_i = \gamma_{i+1}$. At the level of the network, this is accomplished using the matrix $\Gamma$ as

$$\Psi = [\psi_1^T, \psi_2^T, \ldots, \psi_p^T]^T = \Gamma^T \left[ \begin{array}{c}
\gamma_1^T, \gamma_2^T, \ldots, \gamma_p^T
\end{array} \right]^T = \Gamma^T \gamma.$$
With updated information, the local controllers solve the distributed problems separately. Then, the upstream and downstream neighbor vectors are updated. The algorithm continues to loop for the specified number of iterations, and then control actions are performed on local plants. The index $j$ is used for counting the iterations.

B. Convergence of the LC-DMPC Algorithm

**Theorem 1:** For the unconstrained local MPCs, Algorithm 1 is converging, as long as the eigenvalues of matrix $\Delta$ in (20), shown at the bottom of the next page, are all inside the unit circle.

**Proof:** In Algorithm 1, as the local controllers communicate, the vectors $U$, $V$, and $\Psi$ change in the same way as a set of states in any dynamical system. Therefore, the evolution of these vectors is represented as a linear discrete control system. From Step 3 in Algorithm 1, the network control actions are

$$U(j + 1) = \beta U(j) + (1 - \beta) U^{QP}(j)$$

(15)

where

$$U^{QP} = [U_1^{QP} T U_2^{QP} \ldots U_p^{QP} ]^T$$

and from Step 1 and (9) the network disturbance input is

$$V(j + 1) = \Gamma [F_X X_0(k) + M_U U(j + 1) + N_U V(j)].$$

(16)

Similarly, Step 1 and (14) (at the network level), give

$$\Gamma [F_X X_0(k) + M_U U(j + 1) + N_U V(j)].$$

(17)

Now, (15) and (16) can be realized as a linear discrete state space

$$\begin{bmatrix} U(j + 1) \\ V(j + 1) \\ \Psi(j + 1) \end{bmatrix} = \begin{bmatrix} \beta I & 0 & 0 \\ \Gamma M_0 \beta & \Gamma N_z & 0 \\ 2 \Gamma \Gamma T \Psi_{MQ} \beta & 2 \Gamma \Gamma T \Psi_{QM} & \Gamma \Gamma T \Psi \end{bmatrix} \begin{bmatrix} U(j) \\ V(j) \\ \Psi(j) \end{bmatrix} + \begin{bmatrix} (1 - \beta) I \\ (1 - \beta) \Gamma M_z \beta & \Gamma F_z & 0 \\ 2 (1 - \beta) \Gamma \Gamma T \Psi_{MQ} M_y & 2 \Gamma \Gamma T \Psi_{QM} & -2 \Gamma \Gamma T \Psi \end{bmatrix} \begin{bmatrix} U^{QP}(j) \\ X_0(k) \\ r(k) \end{bmatrix}.$$  

(18)

For the unconstrained case, the optimal solution of (13) at iteration $j$ is given by

$$U_{i}^{QP}(j) = [S_i + M_{i,j}^{T} Q_i M_{i,j}^{-1}]^{-1}$$

$$\times [M_{i,j}^{T} Q_i r_i(k) - 0.5 M_{i,j}^{T} \Psi_i(j)$$

$$- M_{i,j}^{T} Q_i N_i X_0(j) - M_{i,j}^{T} Q_i F_i, x_{0,i}(k)].$$

(19)

By stacking all local solutions, the network solution would be

$$U^{QP}(j) = [S + M_{y,j}^{T} Q M_{y,j}^{-1}]^{-1}$$

$$\times [M_{y,j}^{T} Q r(j) - 0.5 M_{y,j}^{T} \Psi(j)$$

$$- M_{y,j}^{T} Q N_j V(j) - M_{y,j}^{T} Q F_j X_0(k)].$$

(19)

and by substituting (19) into (18), the new state matrix is determined as shown in (20), with $X = [S + M_{y,j}^{T} Q M_{y,j}^{-1}]^{-1}$. Stability of (20) the convergence of Algorithm 1. To converge, the eigenvalues of (20) have to be inside the circle. Vectors $X_0$ and $r(k)$ appear as disturbance signals in the new state-space form.

A similar result is also given in [15] for the steady-state case and [16]. For the constrained local MPCs, the convergence of Algorithm 1 is also ensured through the stability of matrix (20) where the free (design) variables are $\beta, Q$, and $S$. It is sufficient only to check the eigenvalues of (20) without solving any additional systemwide problems for the convergence of Algorithm 1.

C. Closed-Loop Stability With the LC-DMPC Algorithm

**Theorem 2:** states the closed-loop stability of Algorithm 1 with the following assumptions.

**Assumption I:** Algorithm 1 is stable, i.e., $|\lambda(\Delta)| < 1$.

**Assumption II:** Algorithm 1 has sufficient iterations per sampling so that (18) converges to a steady-state point.

**Assumption III:** Each pair $(A_i, B_{i,u})$ is controllable $\forall \Sigma_i \in p$.

**Theorem 2:** Under Assumptions I–III, and for a feasible initial condition $x_{0,j}(k)$, the local closed-loop subsystem $\Sigma_i$ with Algorithm 1 is stable for a sufficiently long horizon.

**Proof:** The proof is divided into two parts. The first part proves that Algorithm 1 converges to the centralized solution. The second part displays the stabilization property of this solution for sufficiently long horizons.

**Part 1:** The Karush–Kuhn–Tucker (KKT) conditions are the first-order necessary optimality conditions; thus, if $(\ell)$ is the Lagrange multiplier for (11), then the Lagrangian is given as

$$L(\ell) = U^T H U + 2U^T F - \ell^T (A_{iq} U - B_{iq})$$

and the KKT conditions with systemwide dynamics are

$$2[S + (M_y + N_y WM_z) T Q(M_y + N_y WM_z)] U^{QP}$$

$$+ 2[M_y + N_y WM_z T Q(F_y + N_y WF_z X_0(k))]$$

$$- (M_y + N_y WM_z T Q r(k)) - \ell^T A_{iq} U - B_{iq} = 0$$

(21a)

$$A_{iq} U^{QP} - B_{iq} \leq 0$$

(21b)

$$\ell^* \geq 0$$

(21c)

$$\ell^* (A_{iq} U^{QP} - B_{iq}) = 0$$

(21d)

where the subscript $j$ denotes the $j$th row (active constraints). Also, let $\ell_i$ be the Lagrange multiplier for (13), then

$$L_i(U_i, \ell_i) = U_i^T H_i U_i + 2U_i^T F_i + V_i N_i V_i + 2V_i^T T_i$$

$$- \ell_i^T (A_{iq_i} U_i - B_{iq_i})$$

which, with local dynamics, gives the following KKT conditions:

$$2[S_i + M_{y,i} T Q_i M_{y,i}] U_i^{QP}$$

where

$$U_i^{QP}(j) = [S_i + M_{y,i} T Q_i M_{y,i}]^{-1}$$

$$\times [M_{y,i} T Q_i r_i(k) - 0.5 M_{y,i} T \Psi_i(j)$$

$$- M_{y,i} T Q_i N_i X_0(j) - M_{y,i} T Q_i F_i, x_{0,i}(k)].$$
By stacking conditions (22) for all subsystems, we would have

\[ 2 \left[ S + M_i^T Q M_i \right] U^{OP} \]

\[ + 2 \left[ M_i^T Q F_i x_0(k) + 0.5M_i^T \Psi_i + M_i^T Q N_i V - M_i^T Q r_i(k) \right] \]

\[- \ell_i^T A_{iq} = 0 \] (22a)

\[ A_{iq} U^{OP}_i - B_{iq} \leq 0 \] (22b)

\[ \ell_i^T \geq 0 \] (22c)

\[ \ell_j \left( A_{iq}^T U^{OP}_j - B_{iq,j} \right) = 0 \] (22d)

By substituting (26) into (22e) and rearranging some terms, the final result would be

\[ 2 \left[ S + (M_k + N_k WM_k)^T Q (M_k + N_k WM_k) \right] U^{OP} \]

\[ + 2 \left[ (M_k + N_k WM_k)^T Q (F_k + N_k WF_k) x_0(k) \right] \]

\[- \left( M_k + N_k WM_k \right)^T Q r(k) \] \(- \ell^T A_{iq} = 0 \] (27)

Equation (27) is the same as (21a), i.e., the solution of Algorithm 1 converges to the centralized optimum solution.

**Part 2:** In this part, two theorems detailed in [17] will be restated briefly. Reference [17, Ths. 5.2 and 5.3] implies that for sufficiently large horizons, the solution applied by the centralized MPC in a receding horizon fashion is stabilizing. The idea is to serve the cost in (10) as a Lyapunov function. To accomplish this, Assumption III must be satisfied in addition to the observability of \( (q_i^{1/2}, A_i) \varphi \Sigma_i \in p \) which is already satisfied through the assumptions of \( q_i > 0 \) and \( s_i > 0 \).

As the LC-DMPMC algorithm under Assumptions I and II converges to the same centralized MPC optimum, then it also stabilizes the network for sufficiently large horizons. The recursive feasibility of Algorithm 1 is not studied in this brief and will be a part of future works.

### IV. Laguerre Functions for the Local Problems

In Theorem 2, the closed-loop stability is proven assuming an appropriately long horizon for the centralized problem. This assumption is distributed for the local problems as well. In this brief, Laguerre functions are used to approximate and reduce the size of the distributed problems by parameterizing the local control actions via orthogonal Laguerre networks with \( a \) as a pole for these functions. Many of the following equations are taken from [18]. With Laguerre functions, the local control actions can be reformulated as

\[ u_i(k + m) = \sum_{j=1}^{N_i} c_{ij}(k) \eta_j(m) \] (28)

where \( m = 0, 1, \ldots, N_p \) and \( c_{ij}, j = 1, 2, \ldots, N_i \) are coefficients to be computed. \( l_j(m) \) is the set of Laguerre functions in discrete format. The \( z \) transform of these functions is given by

\[ L_{N_i}(z, a_i) = \frac{(1-a_i^2)z^{-1/2}}{(1-a_i z^{-1})} \left( \frac{z^{-1} - a_i}{1-a_i z^{-1}} \right)^{N_i-1}, \quad 0 \leq a_i < 1 \]

thus, the state-space model of the Laguerre sequences will be

\[ L_i(k + 1) = A_{L_i} L_i \]

where

\[ L_i(k) = [l_1(k), l_2(k), \ldots, l_{N_i}(k)]^T, \quad \beta_i = 1 - a_i^2 \]

\[ A_{L_i} = \begin{bmatrix} a_i & 0 & 0 & \cdots & 0 \\ 0 & \beta_i & a_i & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_i & \cdots & 0 & a_i \end{bmatrix} \]

with

\[ L(0)^T = \sqrt{p_i} [1, -a_i, a_i^2, -a_i^3, \ldots, (-1)^{N_i-1} a_i^{N_i-1}] \].

With this state-space representation, (28) can be written as

\[ u_i(k + j) = L_i(j)^T \eta_i \] (29)

where \( \eta_i^T = [c_1 c_2 \ldots c_{N_i}] \), then the local dynamics (6) and (7) are expressed with (29) as following:

\[ y_i(k + m) = C_{y,i} A_i^{m-1} x_0(i, k) + \Phi_{y,i} \eta_i + n_{y,i} V_i(m) \]

\[ z_i(k + m) = C_{z,i} A_i^{m-1} x_0(i, k) + \Phi_{z,i} \eta_i + n_{z,i} V_i(m) + t_i \] (30)

where

\[ \Phi_{y,i} = \sum_{i=0}^{m-1} C_{y,i} A_i^{m-i-1} B_{a,i} L_i(i)^T \]

\[ n_{y,i} = \sum_{i=0}^{m-1} C_{y,i} A_i^{m-i-1} B_{n,i}, \quad t_i = D_{z,i} L_i(m-1)^T \eta_i \]

\[ V_i(m) = [v_i(0) \ ET \ (v_i(m-1)) \cdots ] \] and \( \Phi_{z,i} \) and \( n_{z,i} \) are defined in a similar way with \( C_{z,i} \) replacing \( C_{y,i} \) with one step late. By substituting (30) into (13)
and using the orthogonality properties of Laguerre functions and $N_c$ as an upper summation limit, the new local problem with reduced manipulated variables is given by

$$J_i = \eta_{p,i}^T \left\{ \sum_{m=1}^{N_p} \left( \Phi_{y,i}^T Q_i \Phi_{y,i} \right) + S_i \right\} \eta_{p,i}$$

$$+ 2\eta_{p,i}^T \left\{ \sum_{m=1}^{N_p} \left( \Phi_{y,i}^T Q_i C_i A_j^m x_{0,i}(k) + \Phi_{y,i}^T Q_i \eta_{y,i} V_i(m) \right) \right\} + 0.5 \Phi_{y,i}^T \psi(m) - \Phi_{y,i}^T \psi_i(k) \right\} + c_0$$

$$s.t. \left[ L_{p,i}(m) \right]^T \eta_{p,i} \leq \left[ -\mu_{\text{max}} \right. , \quad m = 0, 1, \ldots, N_c \right.$$ where $N_c$ is the horizon length at which the control actions are saturated, $\psi(m)$ is the shared sensitivity value at time $m$, and $c_0$ is a constant.

Because of the exponential decay properties of Laguerre functions, the control actions will be faster in the beginning of the horizon and avoid peaks at the end of the horizon [18]. This reduces the number of constraints for the local MPC. The new optimization problem (31) will replace the one given in Step 2 in Algorithm 1 with $\eta_{p,i}$ as new variables. This new problem has smaller decision variables as the number of terms used to parameterize the control actions is the new decision variable.

V. IMPLEMENTATION OF THE LC-DMPC ALGORITHM

In this section, the demonstration of the LC-DMPC approach is given by applying the approach to control a coupled tank process. The section begins with process description, model derivations, and realizations of some matrices. Section V-B presents the simulation results.

A. Six-Tank Process

The control system example used in simulation is a multi-input, multi-output process with six coupled water tanks. A schematic of the process is shown in Fig. 2(a). The objective is to regulate water levels in the lower tanks through three pumps. Pump $q_1$ pours water to tanks 1 and 5, while pump $q_2$ pours water to tanks 2, 4, and 6. Finally, tanks 3 and 5 get water from pump $q_3$. In addition to the coupling in control actions, the outputs from upper tanks likewise disturb the lower tanks. The process has seven valves $V_i$, $i = 1, 2, \ldots, 7$ that control the water flow rates. This system has three control inputs ($v_1, v_2, v_3$), input voltages to pumps, and three outputs ($y_1, y_2, y_3$), water levels in tanks 1, 2, and 3. Due to the strong coupling between the tanks, the task of controlling water levels in the lower tanks is rather difficult to fulfill. The pump flows along with the level of the tanks are constrained physically. However, in this brief, only hard constraints on pump actions are considered. Using mass balances and Bernoulli’s law, the following differential equations can approximate the dynamic of the process:

$$h_i = \frac{q_i(t)}{A_i} + \frac{q_4(t)}{A_1} + \frac{q_3(t)}{A_1} + \frac{q_3(t)}{A_{1}} + \frac{r_1 F_1}{A_1}$$

$$h_2 = \frac{q_2(t)}{A_2} + \frac{q_4(t)}{A_2} + \frac{q_5(t)}{A_2} + \frac{q_5(t)}{A_2} + \frac{q_6(t)}{A_2} + \frac{q_6(t)}{A_2} + \frac{r_2 F_2}{A_2}$$

$$h_3 = \frac{q_3(t)}{A_3} + \frac{q_5(t)}{A_3} + \frac{q_6(t)}{A_3} + \frac{q_6(t)}{A_3} + \frac{r_3 F_3}{A_3}$$

$$h_4 = -\frac{q_4(t)}{A_4} + \frac{q_4(t)}{A_4} + \frac{r_2 F_2}{A_4}$$

$$h_5 = -\frac{q_5(t)}{A_5} + \frac{q_5(t)}{A_5} + \frac{r_2 F_2}{A_5}$$

$$h_6 = -\frac{q_6(t)}{A_6} + \frac{q_6(t)}{A_6} + \frac{r_3 F_3}{A_6}$$

where $q_i(t) = b_i(2gh_i(t))^{1/2}, i = 1, 2, \ldots, 6$ are the inlet and outlet flow rates and $F_j = k_j v_j$, with $j = 1, 2, 3$ are the controlled inlet flow rates from pumps. $h_i(t), b_i$, and $A_i$ refer to the water level, cross section of the outlet hole, and cross-sectional area of tank $i$, respectively. The flow parameters $a, b, \gamma, \rho, \delta$, and $\mu$ determine the amount of water flow to the corresponding tank, therefore, $\sum_{i=1}^2 x_i = 1$ for $x_i = a_i, p_i, r_i$, and $\delta_i$ and $\sum_{j=1}^3 y_j = 1$ for $y_j = v_j$, and $j = 1, 2, 3$. Most of the parameters used in simulation are taken from [19]. The values of the pump voltage constants and flow parameters depend on the operating point. To test Algorithm 1 with the six-tank process, linear models have to be derived. With following deviation variables, the nonlinear model above is linearized around the operating point given in Table I:

$$x_i = h_i - h_i^0, \quad i = 1, 2, \ldots, 6, \quad u_i = v_j - v_j^0, \quad j = 1, 2, 3$$

Using the zero-order hold method, the linear continuous model is discretized with a sampling time of $T_s = 5$ s. This new discretized model is used to implement the centralized
and distributed controllers. In Section II, the subsystems for the tank-process control and input and output disturbances are defined. The following explains how the coupling matrices are defined using subsystem $\Sigma_2$ as an example.

The LC-DMPC definition of subsystem $\Sigma_2$ is given by

$$
x_{S_2}(k+1) = A_{2}x_{S_2}(k) + B_{u,2}u_2(k) + B_{v,2}v_2(k)
$$

$$
y_2(k) = C_{y,2}x_{S_2}(k)
$$

$$
z_2(k) = C_{z,2}x_{S_2}(k) + D_{z,2}u_2(k)
$$

where $x_{S_2} = [x_2 \ x_3]^T$, $v_2(k) = [(x_4(k) u_1(k)) (x_6(k) u_3(k))]^T$, and $A_2, B_{u,2}, B_{v,2}$ and $C_{y,2}$ are derived from the discretized version of the centralized model while the regulated output is $x_2(k)$. This subsystem affects its downstream neighbors $\Sigma_1$ and $\Sigma_3$ through state $x_5(k)$ and control action $u_2(k)$, therefore

$$
z_2(k) = [(x_5(k) \ u_2(k)) (x_3(k) \ u_2(k))]^T.
$$

Then, matrices $C_{z,2}$ and $D_{z,2}$ can be realized as

$$
C_{z,2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad D_{z,1} = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^T.
$$

Following the same procedure, the coupling matrices for subsystems $\Sigma_1$ and $\Sigma_3$ would be:

$$
C_{z,1} = C_{z,3} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad D_{z,1} = D_{z,3} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.
$$

As no direct coupling between subsystems $\Sigma_1$ and $\Sigma_3$ exists, then, according to LC-DMPC architecture, only subsystems $\Sigma_1$ and $\Sigma_2$ and subsystems $\Sigma_2$ and $\Sigma_3$ are sharing data (see Fig. 2(b)).

**B. Simulation Results**

For the six-tank process, Algorithm 1 is simulated and compared with a centralized MPC. For both control structures, the same values for weights, actuator saturations, and prediction horizons are used. The values are: cost weights: $q_1 = 10$, $q_2 = 15$, $q_3 = 14$, $s_1 = 0.1$, $s_2 = 1$, and $s_3 = 0.8$, and constraints on control actions: $|u_1| \leq 2$, $|u_2| \leq 1.8$, and $|u_3| \leq 1.5$. The Laguerre function parameters are $N_j = 20$, $a_j = 0.8$ for $u_i$, $i = 1, 2, 3$. Prediction and constraint horizons are assumed to be $N_p = 100$ and $N_c = 25$, respectively. Finally, the convex combination scalar is $\beta = 0.5$. Two different numbers of iterations are simulated, $N_a = 4$ and $N_a = 1$. Figs. 4 and 5 show the time responses and pump control responses, respectively.

The local controllers can track the centralized solution successfully for $N_a = 4$ through solving a reduced size local optimal control problems. For $N_a = 1$, the local solutions still can converge to the centralized MPC solution. Finally, the eigenvalues of the convergence matrix (20) with the coupled-tank process lie inside the unit circle as shown in Fig. 6, which indicates the stability (convergence) of the algorithm.

**VI. CONCLUSION AND FUTURE WORK**

In this brief, a DMPC algorithm is proposed for coupled and discrete-time linear systems in networks based on neighbor-to-neighbor information sharing structure. The partition of the subsystems is accomplished using an upstream and downstream scheme. Each subsystem requires sharing its predicted effects and sensitivities with its downstream and upstream neighbors, respectively. With local information, the algorithm can converge to the centralized solution through neighbors’
cooperation only, which gives the algorithm a high level of modularity. With local controllability and observability, convergence of the algorithm, and sufficiently long horizons, the closed-loop stability is proven. The local control variables have been parametrized by Laguerre functions, which resulted in a smaller local problem with reduced constraint horizons.

For future work, the iterative feasibility of the local MPCs will be investigated. Also, dissipativity theory will be used to distribute the convergence condition among the subsystems.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NIST or the U.S. Department of Commerce.

REFERENCES


