

Reconfigurable Plug-and-play Distributed Model Predictive Control for Reference Tracking

Conference Paper**Author(s):**

Aboudonia, Ahmed; Martinelli, Andrea ; Hoischen, Nicolas ; Lygeros, John 

Publication date:

2022

Permanent link:

<https://doi.org/10.3929/ethz-b-000591436>

Rights / license:

In Copyright - Non-Commercial Use Permitted

Funding acknowledgement:

180545 - NCCR Automation (phase I) (SNF)
787845 - Optimal control at large (EC)

Reconfigurable Plug-and-play Distributed Model Predictive Control for Reference Tracking

Ahmed Aboudonia, Andrea Martinelli, Nicolas Hoischen and John Lygeros

Abstract—A plug-and-play model predictive control (PnP MPC) scheme is proposed for varying-topology networks to track piecewise constant references. The proposed scheme allows subsystems to occasionally join and leave the network while preserving asymptotic stability and recursive feasibility and comprises two main phases. In the redesign phase, passivity-based control is used to ensure that asymptotic stability of the network is preserved. In the transition phase, reconfigurable terminal ingredients are used to ensure that the distributed MPC problem is initially feasible after the PnP operation. The efficacy of the proposed scheme is evaluated by applying it to a network of mass-spring-damper systems and comparing it to a benchmark scheme. It is found that the novel redesign phase results in faster PnP operations, whereas the novel transition phase increases flexibility by accepting more requests.

I. INTRODUCTION

Plug-and-play (PnP) control schemes have received considerable attention for the control of varying-topology networks, where agents occasionally join and leave the network [1]. The need to integrate constraints has motivated research into combining Model Predictive Control (MPC) and PnP algorithms for ensuring stability and constraint satisfaction of such networks. In [2], decentralized PnP MPC schemes were developed applicable mainly to weakly coupled networks, as they consider all the dynamic coupling terms as disturbances. For the regulation of strongly coupled networks, distributed PnP MPC schemes were developed in [3], [4] based on a two-phase PnP algorithm. In the *redesign phase*, once a PnP request is received, the MPC ingredients are computed for a specific set of subsystems to ensure stability of the new network. Then, the *transition phase* computes a steady state from which the MPC problem is initially feasible for the new network. The PnP request is approved if the optimization problems of both phases are feasible.

We propose a novel PnP algorithm for a class of networked dynamical systems with Laplacian interconnections. Our scheme is inspired by [3], but addresses piecewise constant reference tracking problems, instead of regulation problems. Unlike [3], the proposed PnP algorithm updates the local terminal sets online based on the work in [5]–[7]. This yields larger feasible regions in the transition phase where more PnP requests can be accommodated and also avoids the

computation of a global terminal set which might require a central coordinator. Furthermore, the redesign phase makes use of passivity theory to update the local terminal costs and controllers following the work in [8]. Hence, each subsystem whose terminal ingredients are updated solves only two optimization problems. This accelerates the PnP operation compared to [3], where an iterative procedure is used in the redesign phase. Unlike [8], we consider a more general class of multiple-input-multiple-output systems whose coupling can be described with more than one Laplacian.

In Section II, we introduce the considered class of systems and recall the distributed MPC scheme developed in [7]. In Section III, we develop the novel PnP algorithm which runs once a PnP request is received. In Section IV, we evaluate the efficacy of the proposed scheme using a network of mass-spring-damper (MSD) systems followed by some concluding remarks in Section V.

II. PROBLEM FORMULATION

We consider networked dynamical systems with a distributed structure which can be decomposed into a set \mathcal{M} of M subsystems. Two subsystems are assumed to be neighbours if they are dynamically coupled. The set of neighbours of the i^{th} subsystem is denoted by \mathcal{N}_i and is always assumed to include the i^{th} subsystem. The notation $\mathcal{N}_i \setminus i$ excludes the i^{th} subsystem from its set of neighbours. The i^{th} subsystem is described using the linear time-invariant dynamics

$$\begin{aligned} x_i(t+1) &= A_i x_i(t) + B_i u_i(t) + \sum_{k=1}^{p_i} F_{i,k} v_{i,k}(t), \\ v_{i,k}(t) &= \sum_{j \in \mathcal{N}_i \setminus i} l_{ij,k} (y_{j,k}(t) - y_{i,k}(t)), \quad y_{i,k}(t) = C_{i,k} x_i(t), \end{aligned} \quad (1)$$

where $t \in \mathbb{Z}_+$ is the time index, $x_i \in \mathbb{R}^{n_i}$, $u_i \in \mathbb{R}^{m_i}$ and $y_i \in \mathbb{R}^{p_i}$ are the state, input and output vectors of the i^{th} subsystem and $y_{i,k}$ refers to the k^{th} output of the i^{th} subsystem. The matrices A_i , B_i , vectors $F_{i,k}$, $C_{i,k}$ and scalars $l_{ij,k}$ are known and have appropriate dimensions. We assume that all subsystems have the same output dimension p (i.e. $p = p_i$ for all $i \in \mathcal{M}$). We also assume that the interconnection due to the k^{th} pair $(v_{i,k}, y_{i,k})$ is described using the graph \mathcal{G}_k with the Laplacian $L_k \in \mathbb{R}^{M \times M}$ whose entries are given by

$$L_{ij,k} = \begin{cases} \sum_{j \in \mathcal{N}_i \setminus i} l_{ij,k}, & i = j, \\ -l_{ij,k}, & i \neq j, j \in \mathcal{N}_i, \\ 0, & i \neq j, j \notin \mathcal{N}_i, \end{cases} \quad (2)$$

Note that if $j \in \mathcal{N}_i$, but the k^{th} output of the j^{th} subsystem does not affect the dynamics of the i^{th} subsystem, then $l_{ij,k}$ is set to zero. The dynamics (1) can be also given by

$$x_i(t+1) = A_{N_i} x_{N_i}(t) + B_i u_i(t), \quad y_i(t) = C_i x_i(t), \quad (3)$$

Research supported by the Swiss National Science Foundation under NCCR Automation and by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme grant agreement OCAL, No. 787845. (Corresponding Author: Ahmed Aboudonia)

The authors are with the Automatic Control Laboratory, Department of Electrical Engineering and Information Technology, ETH Zurich, 8092 Zurich, Switzerland. {ahmedab, andremar, lygeros}@control.ee.ethz.ch and nhoischen@student.ethz.ch

where $x_{N_i} \in \mathbb{R}^{n_i}$ is a state vector comprising the states of the subsystems in the set \mathcal{N}_i and A_{N_i} can be constructed using A_i , $C_{i,k}$, $F_{i,k}$ and $l_{i,j,k}$. The states and inputs of the i^{th} subsystem are constrained to polytopic sets,

$$\begin{aligned} x_i(t) \in \mathcal{X}_i &= \{x_i \in \mathbb{R}^{n_i} : G_i x_i \leq g_i\}, \\ u_i(t) \in \mathcal{U}_i &= \{u_i \in \mathbb{R}^{m_i} : H_i u_i \leq h_i\}, \end{aligned} \quad (4)$$

where the matrices $G_i \in \mathbb{R}^{q_i \times n_i}$, $H_i \in \mathbb{R}^{r_i \times m_i}$ and the vectors $g_i \in \mathbb{R}^{q_i}$, $h_i \in \mathbb{R}^{r_i}$ are known. Note that there are no constraints coupling states or inputs of different subsystems.

Assuming that the i^{th} subsystem is required to track the target point $x_{r_i} \in \mathcal{X}_i$ whose corresponding input is $u_{r_i} \in \mathcal{U}_i$, we choose the cost function $J_i = \sum_{t=0}^{T-1} \{ \|x_i(t) - x_{e_i}\|_{Q_i} + \|u_i(t) - u_{e_i}\|_{R_i} \} + \|x_i(T) - x_{e_i}\|_{P_i} + \|x_{e_i} - x_{r_i}\|_{S_i}$, where T is the prediction horizon and (x_{e_i}, u_{e_i}) is an artificial equilibrium of the i^{th} subsystem. The cost function matrices Q_i , R_i , P_i and S_i are known positive definite matrices of appropriate dimensions and the artificial equilibrium of the i^{th} subsystem is required to satisfy

$$x_{e_i} = A_{N_i} x_{e_{N_i}} + B_i u_{e_i} \in \lambda_i \mathcal{X}_i, \quad u_{e_i} = K_i x_{e_i} + d_i \in \lambda_i \mathcal{U}_i, \quad (5)$$

where $K_i \in \mathbb{R}^{m_i \times n_i}$ and $d_i \in \mathbb{R}^{m_i}$ define the terminal controller $\kappa_i(x_i) = K_i x_i + d_i$ and $\lambda_i \in (0, 1)$. Note that P_i and K_i are computed offline such that $V(x) = \|x\|_P^2$ with $x = [x_1^\top, \dots, x_M^\top]^\top$ and $P = \text{diag}(P_1, \dots, P_M)$ is a Lyapunov function for the overall system. Unlike the constraints (4), the costs of the subsystems are coupled indirectly through the dynamic coupling of the equilibrium encoded in (5). The local variables x_i , u_i and x_{N_i} can be extracted from the global variables $x = [x_1^\top, \dots, x_M^\top]^\top \in \mathbb{R}^{n \times n}$ and $u = [u_1^\top, \dots, u_M^\top]^\top \in \mathbb{R}^{n \times n}$ through the projections

$$x_i = U_i x, \quad x_{N_i} = W_i x, \quad u_i = V_i u, \quad (6)$$

where $U_i \in \{0, 1\}^{n_i \times n}$, $W_i \in \{0, 1\}^{n_{N_i} \times n}$ and $V_i \in \{0, 1\}^{m_i \times m}$ are appropriately constructed.

To ensure asymptotic stability and recursive feasibility, the state $x_i(T)$ is constrained to lie in the ellipsoidal positively-invariant terminal set as follows $x_i \in \mathcal{X}_{f_i} = \{x_i \in \mathbb{R}^{n_i} : (x_i - c_i)^\top P_i (x_i - c_i) \leq \alpha_i^2\}$ where α_i and c_i refer to the size and center of the local terminal set of the i^{th} subsystem. This constraint can be approximated using the Schur Complement and diagonal dominance as

$$\begin{aligned} 2\alpha_i [P_i]_j &\geq \bar{b}_{ij} + \alpha_i \{P_i\}_j \quad \forall j \in \{1, \dots, n_i\}, \\ \alpha_i &\geq \sum_{j=1}^{n_i} \bar{b}_{ij}, \quad -\bar{b}_{ij} \leq (x_{ij} - c_{ij}) \leq \bar{b}_{ij}, \end{aligned} \quad (7)$$

where $\bar{b}_{ij} \in \mathbb{R}^{n_i}$ is a decision variable, $[\cdot]_j$ and $\{\cdot\}_j$ refer to the diagonal element and the summation of absolute values of the elements in the j^{th} row of a matrix, respectively. While the terminal set shape represented by P_i is computed offline, the size α_i and center c_i are considered as decision variables computed online. To ensure the positive invariance of the local terminal sets, we use the constraint in (8) which is given overleaf in single column) and

$$G_i^k c_i + \|G_i^k P_i^{-1/2}\|_2 \alpha_i \leq g_i^k, \quad (9)$$

$$H_i^l K_i c_i + H_i^l d_i + \|H_i^l K_i P_i^{-1/2}\|_2 \alpha_i \leq h_i^l, \quad (10)$$

for all $i \in \mathcal{M}$, $k \in \{1, \dots, q_i\}$ and $l \in \{1, \dots, r_i\}$ where $P_{ij} = W_i U_j^\top P_j U_j W_i^\top$, G_i^k and H_i^l are the k^{th} and l^{th} rows of the matrices G_i and H_i respectively, g_i^k and h_i^l are the k^{th} and l^{th} entries of the vectors g_i and h_i . For the derivation of the constraints (8)-(10), refer to [7]. In summary, the online optimization problem is given by

$$\begin{aligned} \min \sum_{i \in \mathcal{M}} J_i \quad \text{s.t.} \quad & x_i(0) = x_{m_i}, \quad (3) - (5), \quad (7) - (10) \\ & \forall i \in \{1, \dots, M\} \ \& \ t \in \{0, \dots, T\}. \end{aligned} \quad (11)$$

where x_{m_i} is the measured state of the i^{th} subsystem and the decision variables are $x_i(t)$, $u_i(t)$, x_{e_i} , u_{e_i} , α_i , c_i , d_i , b_i , \bar{b}_i and ρ_{ij} for all $t \in \{0, \dots, T\}$, $i \in \mathcal{M}$, $j \in \mathcal{N}_i$, $k \in \{1, \dots, q_i\}$ and $l \in \{1, \dots, r_i\}$. Although the distributed MPC problem (11) requires global information, it is still amenable to distributed optimization techniques such as consensus alternating direction method of multipliers (ADMM) and hence, can be solved in a distributed fashion [9].

If subsystems leave or join, the network topology changes and hence the cost and constraints in (11) also change. Thus, the asymptotic stability of the closed-loop system and the recursive feasibility of the MPC problem are no longer guaranteed. To address this difficulty, we develop below a PnP algorithm which allows changes in the network topology while ensuring asymptotic stability and recursive feasibility.

III. PLUG-AND-PLAY DISTRIBUTED MPC

We discuss the PnP algorithm to be run whenever a set \mathcal{J} of new subsystems and/or a set \mathcal{L} of existing subsystems send PnP requests to the network. After the k^{th} PnP request, the subsystems of the k^{th} network can be partitioned into three non-overlapping sets. These sets are the plugged-in set \mathcal{J} comprising the new subsystems joining the network, the neighbour set \mathcal{Z} including the subsystems belonging to neither \mathcal{J} nor \mathcal{L} and whose set of neighbours contains at least one subsystem in \mathcal{J} or \mathcal{L} and the non-neighbour set \mathcal{O} comprising the subsystems belonging to neither \mathcal{J} nor \mathcal{Z} . Similar to [3], the proposed PnP algorithm comprises two phases: the redesign phase ensuring asymptotic stability of the k^{th} network and the transition phase ensuring recursive feasibility of the corresponding MPC scheme. In the sequel, these two phases are discussed in detail.

A. Redesign Phase

In the redesign phase, the local terminal controllers of the plugged-in set \mathcal{J} are designed. The controllers of the neighbour set \mathcal{Z} are also modified to take into account the plugged-in set \mathcal{J} and the plugged-out set \mathcal{L} . The newly-designed and modified terminal controllers are designed to ensure the asymptotic stability of the k^{th} network in the absence of constraints. The weights of the local stage costs of a specific set of subsystems are then updated to ensure the asymptotic stability of the k^{th} network in the presence of constraints. All parameters and variables mentioned in this section are those of the k^{th} network unless otherwise stated.

First, we start with updating the terminal controllers of the subsystems in \mathcal{J} and \mathcal{Z} to asymptotically stabilize the k^{th}

$$2\alpha_i [P_i^{-1}]_k \geq \alpha_i \{P_i^{-1}\}_k + \sum_{l=1}^{n_i} \left(|A_{N_i} + B_i K_i U_i W_i^\top| \alpha_{N_i} \right)_{kl} + b_{i_k} \quad \forall k \in \{1, \dots, n_i\}, \quad -b_i \leq (A_i + B_i K_i) c_{N_i} + B_i d_i - c_i \leq b_i \quad (8a)$$

$$2 \sum_{j \in \mathcal{N}_i} \lambda_{ij} [P_{ij}]_k \geq \sum_{j \in \mathcal{N}_i} \lambda_{ij} \{P_{ij}\}_k + \sum_{l=1}^{n_i} \left(|A_{N_i} + B_i K_i U_i W_i^\top| \alpha_{N_i} \right)_{kl} \quad \forall k \in \{1, \dots, n_i\}, \quad \alpha_i - \sum_{j \in \mathcal{N}_i} \lambda_{ij} \geq \sum_{l=1}^{n_i} b_{i_l}, \quad (8b)$$

network in the absence of constraints. The terminal dynamics of the subsystems in \mathcal{J} and \mathcal{Z} under the control law $u_i = K_i x_i + d_i$ is given by $x_i^+ = (A_i + B_i K_i) x_i + \sum_{k=1}^{p_i} F_{i,k} v_{i,k} + B_i d_i$. By appropriately shifting the equilibrium point, the dynamics can be expressed as

$$\Delta x_i^+ = (A_i + B_i K_i) \Delta x_i + \sum_{k=1}^{p_i} F_{i,k} \Delta v_{i,k} \quad (12)$$

where $\Delta v_{i,k} = \sum_{j \in \mathcal{N}_i \setminus i} l_{ij,k} (\Delta y_{j,k} - \Delta y_{i,k})$ and $\Delta y_{i,k} = C_{i,k} \Delta x_{i,k}$. In addition to the subsystem outputs $\Delta y_{i,k}$, to exploit passivity we define virtual outputs for each subsystem in \mathcal{J} and \mathcal{Z} as

$$\Delta z_{i,k} = C_{i,k} \Delta x_i + D_{i,k} \Delta v_{i,k} \quad \text{for all } k \in \{1, \dots, p_i\}. \quad (13)$$

where $D_i = \text{diag}(D_{i,1}, \dots, D_{i,p_i})$ is computed below. We also define $\Delta z_i = [\Delta z_{i,1}^\top, \dots, \Delta z_{i,p_i}^\top]^\top$, $\Delta v_i = [\Delta v_{i,1}^\top, \dots, \Delta v_{i,p_i}^\top]^\top$, $F_i = [F_{i,1}, \dots, F_{i,p_i}]$ and $C_i = [C_{i,1}^\top, \dots, C_{i,p_i}^\top]$.

Definition III.1 (I10) *The dynamics in (12) is strictly passive with respect to the pair $(\Delta v_i, \Delta z_i)$ if and only if there exist a storage function $V_i(\Delta x_i) \geq 0$ and a dissipation function $\gamma_i(\Delta x_i) \geq 0$ such that*

$$V_i(\Delta x_i^+) - V_i(\Delta x_i) \leq \sum_{k=1}^{p_i} \Delta v_{i,k} \Delta z_{i,k} - \gamma_i(\Delta x_i) \quad (14)$$

Note that we use virtual outputs because the dynamics in (1) can not be passivated using the actual outputs [8].

Moreover, we define $L = \sum_{k=1}^p L_k \otimes e_k e_k^\top$ where \otimes refers to the Kronecker product and $e_k \in \mathbb{R}^p$ is a unit vector whose k^{th} element equals one. One can show that the matrix $L \geq 0$ since $L_k \geq 0$ and $e_k e_k^\top \geq 0$. Considering $C = \text{diag}(C_1, \dots, C_M)$, we define $L_C = LC$ and $C_L = C^\top L^\top$ which can be decomposed as follows, $L_C = [L_{C_1}^\top, \dots, L_{C_M}^\top]^\top$ and $C_L = [C_{L_1}^\top, \dots, C_{L_M}^\top]^\top$ where $L_{C_i} \in \mathbb{R}^{p_i \times n_i}$ and $C_{L_i} \in \mathbb{R}^{n_i \times p_i}$. Recall that we denote the diagonal element in the j^{th} row of a matrix by $[\cdot]_j$ and the summation of absolute values of the elements in the j^{th} row by $\{\cdot\}_j$. Finally, we define the scalars n_{ij} for all $i, j \in \{1, \dots, M\}$ such that $n_{ij} > 0$ if $i \in \mathcal{N}_j$, $n_{ij} = 0$ otherwise and $\sum_{j=1}^M n_{ij} \leq 1$ for all $i \in \{1, \dots, M\}$. We now make the following assumption on the parameters and variables of the zeroth network (before any PnP operation) which are different from those of the k^{th} network used in the rest of this section; here \mathbb{S}_{++}^n and \mathbb{D}_{++}^n are the sets of $n \times n$ symmetric and diagonal positive definite matrices, respectively.

Assumption III.1 *Before receiving any PnP request, the dynamics of the i^{th} subsystem in the zeroth network under the control law $u_i = K_i x_i + d_i$ is strictly passive with respect to the pair $(\Delta v_i, \Delta z_i)$ with quadratic storage function $V_i(\Delta x_i) = \Delta x_i^\top P_i \Delta x_i$ and dissipation function $\gamma_i(\Delta x_i) = \Delta x_i^\top \Gamma_i \Delta x_i$. The matrices $P_i \in \mathbb{S}_{++}^{n_i}$, $\Gamma_i \in \mathbb{D}_{++}^{n_i}$ and $D_i \in \mathbb{D}_{++}^{n_i}$ satisfy $[\Gamma_i^{-1}]_j \leq \frac{1}{\{L_{C_i}\}_j + \varepsilon_i}$ for all $j \in \{1, \dots, n_i\}$ and $[D_i]_j \leq \frac{1}{\{C_{L_i}\}_j}$ for all $j \in \{1, \dots, p_i\}$ such that $\{C_{L_i}\}_j > 0$ where ε_i are arbitrarily small positive scalars. Finally, the stage cost*

weights $Q_i \in \mathbb{S}_{++}^{n_i}$ and $R_i \in \mathbb{S}_{++}^{m_i}$ of the i^{th} subsystem satisfy $\sum_{j \in \mathcal{N}_i} n_{ij} W_i U_j^\top P_j U_j W_i^\top - W_i U_i^\top (Q_i + K_i^\top R_i K_i) U_i W_i^\top - (A_{N_i} + B_i K_i U_i W_i^\top)^\top P_i (A_{N_i} + B_i K_i U_i W_i^\top) \geq 0$.

Assumption III.1 can be ensured during the offline synthesis of the MPC scheme (11) for the zeroth network. This assumption ensures that the zeroth network is asymptotically stable in the presence of constraints under the MPC scheme (11) and in the absence of constraints under the controller $u_i = K_i x_i + d_i$ for all $i \in \mathcal{M}$ (see [7], [8] for more details).

Theorem III.1 *After the k^{th} PnP request, consider the k^{th} network under the controller $u_i = K_i x_i + d_i$ for all $i \in \{1, \dots, M\}$. Under Assumption III.1, this network is asymptotically stable in the absence of constraints if for each subsystem in \mathcal{J} and \mathcal{Z} of this network, there exist $E_i \in \mathbb{S}_{++}^{n_i}$, $X_i \in \mathbb{D}_{++}^{n_i}$, $Y_i \in \mathbb{R}^{n_i \times m_i}$ and $D_i \in \mathbb{D}_{++}^{n_i}$ such that*

$$\begin{bmatrix} E_i & \frac{1}{2} E_i C_i^\top & (A_i E_i + B_i Y_i)^\top & E_i \\ \frac{1}{2} C_i E_i & D_i & F_i^\top & 0 \\ (A_i E_i + B_i Y_i) & F_i & E_i & 0 \\ E_i & 0 & 0 & X_i \end{bmatrix} \geq 0, \quad (15a)$$

$$[X_i]_j \leq \frac{1}{\{L_{C_i}\}_j + \varepsilon_i} \quad \forall j \in \{1, \dots, n_i\}, \quad (15b)$$

$$[D_i]_j \leq \frac{1}{\{C_{L_i}\}_j} \quad \forall j \in \{1, \dots, p_i\} \quad \text{such that } \{C_{L_i}\}_j > 0, \quad (15c)$$

where ε_i are arbitrarily small positive scalars.

Proof: see Appendix. \blacksquare

Note that if the considered network is asymptotically stable under the controller $u_i = K_i x_i + d_i$ where $d_i = -K_i x_{r_i} + u_{r_i}$, then the network converges to the target point x_{r_i} . We now move to the second step of the redesign phase in which we update the costs of a specific set of subsystems to asymptotically stabilize the k^{th} network in the presence of constraints. For this purpose, we define the matrices $P = \text{diag}(P_1, \dots, P_M)$, $K = \text{diag}(K_1, \dots, K_M)$, $Q = \text{diag}(Q_1, \dots, Q_M)$ and $R = \text{diag}(R_1, \dots, R_M)$. To ensure the asymptotic stability of the k^{th} network in the presence of constraints under the MPC controller [6], the Lyapunov condition $V(x^+) - V(x) \leq -l(x)$ should be satisfied where $V(x) = x^\top P x$ and $l(x) = x^\top (Q + K^\top R K) x$. This, in turn, is satisfied if the inequality $P - (A + BK)^\top P (A + BK) - Q - K^\top R K \geq 0$. This condition can be ensured by appropriately tuning Q and R . Although this inequality requires global information, it can be used to derive local conditions for each subsystem.

Theorem III.2 *After the k^{th} PnP request, consider the k^{th} network under the MPC controller (11). Under Assumption III.1, this network is asymptotically stable in the presence of constraints if for each subsystem in \mathcal{J} , \mathcal{Z} and their*

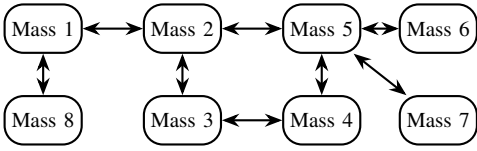


Fig. 1: Network Topology

neighbours, there exist $Q_i \in \mathbb{S}_{++}^{n_i}$ and $R_i \in \mathbb{S}_{++}^{m_i}$ such that

$$\sum_{j \in \mathcal{N}_i} n_{ij} W_i U_j^\top P_j U_j W_i^\top - W_i U_i^\top (Q_i + K_i^\top R_i K_i) U_i W_i^\top - (A_{N_i} + B_i K_i U_i W_i^\top)^\top P_i (A_{N_i} + B_i K_i U_i W_i^\top) \geq 0. \quad (16)$$

Proof: see Appendix. ■

Notice that following [6], asymptotic stability is ensured in Theorem III.2 with respect to the target point x_{r_i} . Note also that each subsystem in \mathcal{J} , \mathcal{Z} , \mathcal{L} and their neighbours can satisfy (16) by solving a constrained optimization problem whose cost function is selected so that the k^{th} network can meet the required performance specifications. The PnP request is rejected if any of these optimization problems is infeasible. Note that the scalars n_{ij} can be also considered as decision variables in such optimization problems and hence the PnP request is rejected if $\sum_{j=1}^M n_{ij} > 1$ for any $i \in \mathcal{M}$.

B. Transition Phase

In the transition phase, we compute for the old network and the new subsystems steady states x_{s_i} at which the PnP operations can take place. These steady states lead to safe PnP operations during which $x_i(t+1) = x_i(t)$ for all subsystems. To compute these steady states, an optimization problem is solved in a distributed manner. In this problem, we ensure that the old network and the new subsystems can reach their steady states starting from their current states before the PnP operation. We also ensure that the terminal sets of the new network and the plugged-out subsystems are reachable from the steady states after the PnP operation. Unlike [3], the terminal sets are considered as decision variable in this optimization problem. In the sequel, the superscript $(\cdot)^o$ refers to the parameters of the old network before the PnP operations, whereas the superscript $(\cdot)^n$ refers to those of the new network after the PnP operations.

The prediction horizon in the transition phase is divided into two periods at $T_{pp} \in [0, T]$. In the first period, we drive the current state of the old network and the new subsystems to the steady state while satisfying the constraints before the PnP operation. For this purpose, we consider the dynamics and constraints of the old network and the new subsystems before joining the network as follows,

$$\forall i \in \mathcal{M}^o \cup \mathcal{J} \quad \begin{cases} x_i(0) = x_{m_i}, & x_i(T_{pp}) = x_{s_i} \\ \forall t \in \{1, \dots, T_{pp} - 1\} \end{cases} \begin{cases} x_i(t+1) = A_{N_i}^o x_{N_i}(t) + B_i^o u_i(t), \\ x_i(t) \in \mathcal{X}_i^o, u_i(t) \in \mathcal{U}_i^o \end{cases} \quad (17)$$

In the second period, we drive the states of the new network and the plugged-out subsystems to their terminal sets while satisfying their dynamics and constraints after the PnP operation. For this purpose, we consider the dynamics

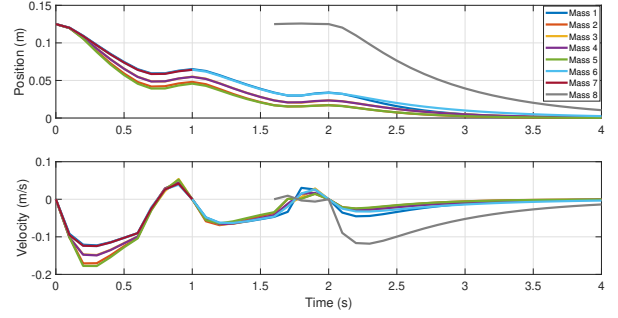


Fig. 2: State trajectories of all masses

and constraints of the new network and the plugged-out subsystems after leaving the network as follows,

$$\forall i \in \mathcal{M}^n \cup \mathcal{L} \quad \begin{cases} x_i(0) = x_{s_i}, & (7) \\ x_i(t+1) = A_{N_i}^n x_{N_i}(t) + B_i^n u_i(t), & (18) \\ \forall t \in \{T_{pp} + 1, \dots, T\} \end{cases} \begin{cases} x_i(t) \in \mathcal{X}_i^n, \\ u_i(t) \in \mathcal{U}_i^n \end{cases}$$

where (7) considers the new network to ensure $x_i(T) \in \mathcal{X}_{f_i}^n$.

To guarantee that the steady states x_{s_i} is a feasible equilibrium point for the old network and the new subsystems before joining the network, we consider the constraints,

$$x_{s_i} = A_i^o x_{s_{N_i}} + B_i^o u_{s_i}, \quad x_{s_i} \in \mathcal{X}_i^o, \quad u_i(T_{pp}) = u_{s_i} \in \mathcal{U}_i^o \quad (19)$$

where u_{s_i} is the input corresponding to the steady state x_{s_i} and $x_{s_{N_i}}$ is a concatenated vector which includes the steady states of the subsystems in the set N_i^o . Finally, we also consider the artificial equilibrium constraints (5) and the terminal set constraints (8)-(10) for the new network. In summary, the transition phase is performed by solving the optimization problem

$$\min \sum_{i \in \mathcal{M}^o \cup \mathcal{J}} \|x_{m_i} - x_{s_i}\| \quad \text{s.t. (17)-(19), (5), (8)-(10), (20)}$$

where the dynamics and constraints of the new network are used in (5) and (7)-(10). If this optimization problem is feasible, the PnP request is accepted and the control sequence $[u_i(0), \dots, u_i(T_{pp})]$ is applied to all subsystems before the PnP operation to drive the whole network to steady state. Once the steady state is reached, the distributed MPC problem (11) is solved recursively for the new network. Similar to the distributed MPC problem (11), the transition phase can be solved in a distributed manner using ADMM [9].

Theorem III.3 *The MPC problem (11) is recursively feasible with the new network starting from the steady state x_{s_i} if the optimization problem (20) is feasible.*

Proof: see Appendix. ■

IV. SIMULATION RESULTS

The efficacy of the proposed PnP algorithm is illustrated using a network of MSD systems. The dynamics of each mass is given in continuous time by $\dot{x}_i = A_{ii} x_i + B_i u_i + F_{i,1} l_{ij,1} (C_{j,1} x_j - C_{i,1} x_i) + F_{i,2} l_{ij,2} (C_{j,2} x_j - C_{i,2} x_i)$ where, for all $i \in \mathcal{M}$, x_i comprises the position and velocity, u_i is the applied force, $B_i = F_{i,1} = F_{i,2} = [0 \ 1]^\top$, $C_{i,1} = [1 \ 0]$, $C_{i,2} = [0 \ 1]$, $l_{ij,1} = k_{ij}$, $l_{ij,2} = c_{ij}$ and $A_{ii} = \begin{bmatrix} 0 & 1 \\ k_i/m_i & c_i/m_i \end{bmatrix}$.

The parameters m_i , k_i and c_i represent the mass, stiffness and damping of mass i and the parameters k_{ij} and c_{ij} represent the stiffness and damping connecting masses i and j where $m_i = k_i = 1$, $c_i = 2$, $k_{ij} = 0.4$ and $c_{ij} = 0.8$ for all $i \in \mathcal{M}$ and $j \in \mathcal{N}_i \setminus i$. All states and inputs are constrained between -1 and 1. The system is discretized using the method in [8], [11] to preserve the structure. The sampling time and prediction horizon are 0.1 seconds and 8 timesteps, respectively. All optimization problems are solved using MATLAB with Yalmip [12] and Mosek [13] on a computer equipped with a 1.9-GHz Intel core i7-8550U processor.

The simulation considers a network of eight MSD systems with the topology in Fig. 1. The simulation starts without the eighth mass. The seventh mass sends a plug-out request at $t = 0.6$ followed by a plug-in request by the eighth mass at $t = 1.6$. Fig. 2 shows the state trajectories of all masses starting from the initial condition $[0.125 \ 0]^T$ towards the target $[0 \ 0]^T$. Note that the network is converging to the target despite the PnP operations. From $t = 0.6$ till $t = 1$ and from $t = 1.6$ till $t = 2$, all subsystems run the transition phase and move to their steady states at which their velocities are zero so that subsystem 7 is plugged out and subsystem 8 is plugged in, respectively. Apart from these two periods, all subsystems run the MPC scheme to converge to the target point. The redesign phase is run in parallel to the MPC scheme before the transition phase.

We compare our algorithm to the one in [3]. First, we compare the redesign phase by running 100 simulations with different coupling strength $k_{18} = 2.5r$ and $c_{18} = 5r$ for all $r \in \{0.01, \dots, 1\}$. We report the number of times n_R the redesign phase is feasible and the mean μ_R and standard deviation σ_R of the time taken in seconds by the redesign phase in Table I. Unlike the algorithm of [3], our algorithm fails for strong couplings, but is much faster. This is because the optimization problem of our algorithm is conservative, but completely decentralized, unlike that of [3] that is distributed and solved using ADMM, requiring hundreds of iterations and hence, is computationally demanding. We also compare the transition phase by running 100 simulations with different initial position $p_8 = 0.5r$ for mass 8 for all $r \in \{0.01, \dots, 1\}$. Both algorithms use ADMM to solve the optimization problem of the transition phase. We report the number of times n_T the transition phase is feasible and the mean μ_T and standard deviation σ_T of the time required in seconds per ADMM step. As shown in Table I, our algorithm requires slightly more time as it has more constraints. But, it is feasible for all initial conditions due to the additional flexibility of computing the terminal sets online.

V. CONCLUSIONS

A PnP algorithm is developed for varying-topology networks tracking piecewise constant references. The algorithm comprises a redesign phase based on passivity-based control and a transition phase that makes use of reconfigurable terminal ingredients. The redesign phase results in faster PnP operations and the transition phase adds more flexibility by accepting more PnP requests compared to a benchmark

TABLE I: The novel algorithm vs benchmark algorithm [3]

Redesign	Existing	New	Transition	Existing	New
n_R	100	82	n_T	15	100
μ_R (s)	29.73	0.008	μ_T (s)	0.0024	0.003
σ_R (s)	3.07	0.0006	σ_T (s)	0.0004	0.0004

scheme. Future work includes using clustering to steer a specific set of subsystems to steady state instead of steering the whole network during PnP operations.

APPENDIX

Proof of Theorem III.1: We use induction to prove the asymptotic stability of the k^{th} network under the controller $u_i = K_i x_i + d_i$ for all $i \in \mathcal{M}$ in the absence of constraints. First, we start with the base case. Based on Assumption III.1, the zeroth network before any PnP request is asymptotically stable. Next, we move to the inductive step. We assume that the $(k-1)^{\text{th}}$ network is asymptotically stable (i.e. satisfies the conditions in Assumption III.1). We derive conditions which ensure that the k^{th} network is asymptotically stable.

First, we derive conditions which ensure that all subsystems of the k^{th} network are strictly passive, or equivalently that (14) holds for all $i \in \mathcal{M}$. Note that this condition is satisfied for \mathcal{O} since the dynamics of such subsystems in the k^{th} network remain the same as in the $(k-1)^{\text{th}}$ network which satisfies Assumption III.1. Hence, it suffices to ensure that (14) holds for \mathcal{J} and \mathcal{Z} . Following [8], we take $V_i(\Delta x_i) = \Delta x_i^T P_i \Delta x_i$ and $\gamma_i(\Delta x_i) = \Delta x_i^T \Gamma_i \Delta x_i$, substitute (12) and (13) in (14), use the Schur Complement twice to reach (15a) where $E_i = P_i^{-1}$, $Y_i = K_i P_i^{-1}$, $X_i = \Gamma_i^{-1}$. The conditions $E_i \in \mathbb{S}_{++}^{n_i}$ and $X_i \in \mathbb{D}_{++}^{n_i}$ ensure that $V_i(x_i) \geq 0$ and $\gamma_i(x_i) \geq 0$.

Second, we derive conditions which ensure that strict passivity of all subsystems in the k^{th} network imply asymptotic stability of this network. For this, we define $\Delta z = [\Delta z_1^T, \dots, \Delta z_M^T]^T$, $\Delta v = [\Delta v_1^T, \dots, \Delta v_M^T]^T$, $C = \text{diag}(C_1, \dots, C_M)$ and $D = \text{diag}(D_1, \dots, D_M)$. We also define the Lyapunov function $V(\Delta x) = \sum_{i=1}^M V_i(\Delta x_i)$ and the auxiliary function $\gamma(\Delta x) = \sum_{i=1}^M \gamma_i(\Delta x_i) = \Delta x^T \Gamma \Delta x$ with $\Gamma = \text{diag}(\Gamma_1, \dots, \Gamma_M)$. Recall that (14) holds for all subsystems in \mathcal{O} with quadratic storage and dissipation functions. Moreover, (15a) implies that (14) holds for all subsystems in \mathcal{J} and \mathcal{Z} with quadratic storage and dissipation functions. By summing up (14) for all $i \in \mathcal{M}$, we reach $V(\Delta x^+) - V(\Delta x) \leq \Delta v^T \Delta z - \gamma(\Delta x)$ where $\Delta z = C \Delta x + D \Delta v$, $\Delta v = -L C \Delta x$ and $\gamma(\Delta x) = \Delta x^T \Gamma \Delta x$.

To prove asymptotic stability of the k^{th} network under the controller $u_i = K_i x_i + d_i$ for all $i \in \{1, \dots, M\}$ in the absence of constraints, it suffices to ensure that $\Delta v^T \Delta z - \gamma(\Delta x) < 0$ or equivalently, $\Gamma + C^T L C - C^T L^T D L C \geq \text{diag}(\varepsilon_1 I_{n_1}, \dots, \varepsilon_M I_{n_M})$ where ε_i are arbitrarily small positive scalars and I_{n_i} is an $n_i \times n_i$ identity matrix. Following [8], we exploit the Schur Complement, diagonal dominance and the fact that $L \geq 0$, $X_i \in \mathbb{D}_{++}^{n_i}$ and $D_i \in \mathbb{D}_{++}^{n_i}$ to reach (15b)-(15c) for all subsystems in \mathcal{J} , \mathcal{Z} and \mathcal{O} . Note that (15b)-(15c) are satisfied for all subsystems in \mathcal{O} since the neighbours of these subsystems remain the same and hence, the matrices C_{L_i} and L_{C_i} do not change. Hence, it suffices to ensure that (15b)-(15c) are satisfied for all subsystems in \mathcal{J} and \mathcal{Z} .

If (15) is not satisfied for one or more subsystems, the PnP request is rejected. The network remains the same and, hence asymptotic stability is preserved and the conditions in Assumption III.1 hold until the next PnP request. If (15) is satisfied for all subsystems, the PnP request can be accepted depending on the next steps in the PnP algorithm. If it is, the controllers are redesigned based on (15) and the conditions in Assumption III.1 hold after the PnP operation is completed, hence until the next PnP request.

Proof of Theorem III.2: Using induction, we prove the asymptotic stability of the k^{th} network under the MPC controller (11) in the presence of constraints. First, we start with the base case. Based on Assumption III.1, the zeroth network before any PnP request is asymptotically stable. Next, we move to the inductive step. We assume that the $(k-1)^{\text{th}}$ network is asymptotically stable (i.e. satisfies the conditions in Assumption III.1). We derive conditions which ensure that the k^{th} network is asymptotically stable.

According to [6], the k^{th} network under (11) is asymptotically stable if the global inequality $P - (A + BK)^{\top}P(A + BK) - Q - K^{\top}RK \geq 0$ holds. This inequality holds if and only if $x^{\top}(P - (A + BK)^{\top}P(A + BK) - Q - K^{\top}RK)x \geq 0$ for all $x \in \mathbb{R}^n$. Due to the imposed structure on the matrices P_i , K_i , Q_i and R_i , this inequality can be written as $\sum_{i=1}^M x_i^{\top} P_i x_i - \sum_{i=1}^M (A_{N_i} x_{N_i} + B_i K_i x_i)^{\top} P_i (A_{N_i} x_{N_i} + B_i K_i x_i) - \sum_{i=1}^M x_i^{\top} (Q_i + K_i^{\top} R_i K_i) x_i \geq 0$ or equivalently,

$$\begin{aligned} & \sum_{i=1}^M \left(1 - \sum_{j=1}^M n_{ij}\right) x_i^{\top} P_i x_i + \sum_{i=1}^M \left(\sum_{j=1}^M n_{ij}\right) x_i^{\top} P_i x_i \\ & - \sum_{i=1}^M (A_{N_i} x_{N_i} + B_i K_i x_i)^{\top} P_i (A_{N_i} x_{N_i} + B_i K_i x_i) \\ & - \sum_{i=1}^M x_i^{\top} (Q_i + K_i^{\top} R_i K_i) x_i \geq 0. \end{aligned}$$

Since $\sum_{j=1}^M n_{ij} \leq 1$ by design, it suffices to ensure that $\sum_{i=1}^M \left(\sum_{j=1}^M n_{ij}\right) x_i^{\top} P_i x_i - \sum_{i=1}^M (A_{N_i} x_{N_i} + B_i K_i x_i)^{\top} P_i (A_{N_i} x_{N_i} + B_i K_i x_i) - \sum_{i=1}^M x_i^{\top} (Q_i + K_i^{\top} R_i K_i) x_i \geq 0$. Using (6), the resulting inequality can be reformulated as

$$\begin{aligned} & \sum_{i=1}^M \left(\sum_{j=1}^M n_{ij}\right) x_{N_i}^{\top} W_i U_i^{\top} P_i U_i W_i^{\top} x_{N_i} \\ & - \sum_{i=1}^M x_{N_i}^{\top} (A_{N_i} + B_i K_i U_i W_i^{\top})^{\top} P_i (A_{N_i} + B_i K_i U_i W_i^{\top}) x_{N_i} \\ & - \sum_{i=1}^M x_{N_i}^{\top} W_i U_i^{\top} (Q_i + K_i^{\top} R_i K_i) U_i W_i^{\top} x_{N_i} \geq 0. \end{aligned}$$

One can show that this global inequality can be decomposed into the local inequalities $\sum_{j=1}^M n_{ij} x_{N_i}^{\top} W_i U_j^{\top} P_j U_j W_i^{\top} x_{N_i} - x_{N_i}^{\top} (A_{N_i} + B_i K_i U_i W_i^{\top})^{\top} P_i (A_{N_i} + B_i K_i U_i W_i^{\top}) x_{N_i} - x_{N_i}^{\top} W_i U_i^{\top} (Q_i + K_i^{\top} R_i K_i) U_i W_i^{\top} x_{N_i} \geq 0$ for all $i \in \mathcal{M}$. Since $n_{ij} = 0$ if i and j are not neighbours, this holds if and only if the local LMIs (16) are satisfied for all $i \in \mathcal{M}$. Note that these LMIs are satisfied for all subsystems excluding \mathcal{J} , \mathcal{Z} and their neighbours since the $(k-1)^{\text{th}}$ network satisfies Assumption III.1. Hence, it suffices to ensure them for these subsystems.

If (16) is not satisfied for one or more subsystems, the PnP request is rejected. The network remains the same and, hence asymptotic stability is preserved and the conditions in Assumption III.1 hold until the $(k+1)^{\text{th}}$ PnP request. If (16) is satisfied for all subsystems, the PnP request can be

accepted depending on the next steps in the PnP algorithm. If it is, the stage cost weights are redesigned according to (16) and the conditions in Assumption III.1 hold after the PnP operation is completed, hence until the next PnP request.

Proof of Theorem III.3: Let $(x_i(0), \dots, x_i(T))$ and $(u_i(0), \dots, u_i(T-1))$ be feasible trajectories of the optimization problem (20) for the i^{th} subsystem. Therefore, each subsystem can be steered to its steady state x_{s_i} in finite time since $x_i(T_{pp}) = x_{s_i}$ and $u_i(T_{pp}) = u_{s_i}$. Moreover, let α_i and c_i refer to feasible size and center of the terminal set computed by (20). Define $x_{i,tail} = (x_{i,T+1}, \dots, x_{i,T+T_{pp}})$ where $x_{i,j} = (A_{N_i}^n + B_i^n K_i^n U_i^n W_i^{n\top})^{j-T} x_{N_i}(T) + \sum_{k=1}^{j-T} (A_{N_i}^n + B_i^n K_i^n U_i^n W_i^{n\top})^{k-1} B_i^n d_i$ and $u_{i,tail} = (K_i x_i(T) + d_i, K_i x_{i,T+1} + d_i, \dots, K_i x_{i,T+T_{pp}-1} + d_i)$. Note that $x_{i,tail}$ is the evolution of the terminal dynamics and hence lies inside the terminal set described by α_i and c_i . Thus, $(x_i(T_{pp}), \dots, x_i(T), x_{i,tail})$, $(u_i(T_{pp}), \dots, u_i(T-1), u_{i,tail})$, α_i and c_i represent a feasible solution for the MPC problem (11) applied to the new network. In other words, the MPC scheme for the new network is initially feasible. Recursive feasibility can thus be ensured using standard MPC arguments [14].

REFERENCES

- [1] J. Stoustrup, "Plug & play control: Control technology towards new challenges," *European Journal of Control*, vol. 15, no. 3-4, pp. 311–330, 2009.
- [2] S. Rivero, M. Farina, and G. Ferrari-Trecate, "Plug-and-play model predictive control based on robust control invariant sets," *Automatica*, vol. 50, no. 8, pp. 2179–2186, 2014.
- [3] M. N. Zeilinger, Y. Pu, S. Rivero, G. Ferrari-Trecate, and C. N. Jones, "Plug and play distributed model predictive control based on distributed invariance and optimization," in *52nd IEEE conference on decision and control*, pp. 5770–5776, IEEE, 2013.
- [4] H. Hu, Y. Pu, M. Chen, and C. J. Tomlin, "Plug and play distributed model predictive control for heavy duty vehicle platooning and interaction with passenger vehicles," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 2803–2809, IEEE, 2018.
- [5] A. Aboudonia, A. Eichler, and J. Lygeros, "Distributed model predictive control with asymmetric adaptive terminal sets for the regulation of large-scale systems," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6899–6904, 2020.
- [6] A. Aboudonia, A. Eichler, F. Cordiano, G. Banjac, and J. Lygeros, "Distributed model predictive control with reconfigurable terminal ingredients for reference tracking," *IEEE Transactions on Automatic Control*, 2021.
- [7] A. Aboudonia, G. Banjac, A. Eichler, and J. Lygeros, "Online computation of terminal ingredients in distributed model predictive control for reference tracking," in *2022 European Control Conference (ECC)*, IEEE, 2022 (Accepted).
- [8] A. Aboudonia, A. Martinelli, and J. Lygeros, "Passivity-based decentralized control for discrete-time large-scale systems," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2072–2077, 2020.
- [9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," vol. 3(1), pp. 1–122, 2011.
- [10] M. Aliyu, *Nonlinear H-control, Hamiltonian systems and Hamilton-Jacobi equations*. CRC, 2011.
- [11] S. Rivero, M. Farina, and G. Ferrari-Trecate, "Plug-and-play decentralized model predictive control for linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2608–2614, 2013.
- [12] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, (Taipei, Taiwan), 2004.
- [13] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019.
- [14] B. Kouvaritakis and M. Cannon, "Model predictive control," *Switzerland: Springer International Publishing*, 2016.