Computers and Chemical Engineering 155 (2021) 107532

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Coordinating distributed MPC efficiently on a plantwide scale: The Lyapunov envelope algorithm

Wentao Tang [a,b], Prodromos Daoutidis [a,*]

[a] Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455, USA
[b] Surface Operations, Projects and Technology, Shell Global Solutions (U.S.) Inc., Houston, TX 77082, USA

ARTICLE INFO

Article history:
Received 15 April 2021
Revised 3 July 2021
Accepted 7 September 2021
Available online 8 September 2021

Keywords:
Distributed optimization
Model predictive control
Early termination
Lyapunov envelope

ABSTRACT

The model predictive control (MPC) of large-scale systems should adopt a distributed optimization approach, where controllers for the constituent subsystems optimize their control actions and iterations are used to coordinate their decisions. The real-time implementation of MPC, however, usually allows very limited time for computation and inevitably needs to be terminated early. In this work, we propose a splitting algorithm for distributed optimization analogous to forward-backward splitting (FBS), where ℓ_1 and quadratic penalties are imposed on the violation of interconnecting relations among the subsystems. By designing the involved parameters based on dissipative analysis, the iterations result in the monotonic decrease of a plant-wide Lyapunov function, which we call Lyapunov envelope, thus maintaining closed-loop stability under distributed MPC despite early termination and yielding improving control performance as the allowed computational time or number of iterations increases. The proposed Lyapunov envelope algorithm is tested on an industrial-scale vinyl acetate monomer process.

1. Introduction

The need for adopting decompositions in plantwide control has been identified by researchers more than 40 years ago. We quote from Morari et al. (1980):

> "*Decomposition is the underlying, guiding principle, leading to the classification of the control objectives (regulation, optimization) and the partitioning of the process for the practical implementation of the control structures.*"
>
> "*For our purposes, i.e. the systematic and organized development of control structures for chemical processes, we found the framework of the multilayer-multiechelon concept to be very meaningful, convenient, and having the potential for further development.*"

Since the 1980s, a vast volume of literature has been devoted to the synthesis of decentralized plantwide control structure for base-layer loops, i.e., the selection and pairing of manipulated inputs and controlled outputs, see e.g. Stephanopoulos (1983), Price and Georgakis (1993), Luyben et al. (1997), Skogestad (2004). In the contemporary practice of process industries, model predictive control (MPC) has been typically used as the advanced control technology in companion with the base-layer control (Morari and Lee, 1999; Qin and Badgwell, 2003). Currently, the vertical "multilayer" decomposition into a static problem with the objective of optimizing economic cost and satisfying operating constraints and a dynamic problem with the objective of regulating the process at the setpoint is commonly adopted (Rawlings et al., 2017). Another type of hierarchical decomposition that has been proposed is based on the multi-time-scale behavior of plantwide dynamics in the case of tight integration with large material and energy recycles (Kumar et al., 1998; Baldea and Daoutidis, 2007). The horizontal "multi-echelon" decomposition of processes in MPC, frequently referred to as *distributed MPC*, has also been explored in the recent literature (Scattolini, 2009; Christofides et al., 2013; Negenborn and Maestre, 2014). Based on perspectives from modern network science, community detection methods have been proposed to generate high-quality decompositions for distributed MPC (Daoutidis et al., 2018; 2019).

Broadly speaking, distributed MPC may refer to any such iterative algorithms where during the iterations, the subsystem controllers mutually share some information that helps to obtain a solution better than the decentralized one (e.g., see Negenborn and Maestre, 2014 for a comprehensive review). In a narrower sense, distributed MPC implies the adoption of *distributed optimization* algorithms (Yang et al., 2019), whose iterations are not simply informative (passing information from one agent to another) but also coordinative (aiming at approaching solutions of the entire sys-

* Corresponding author.
E-mail address: daout001@umn.edu (P. Daoutidis).

https://doi.org/10.1016/j.compchemeng.2021.107532
0098-1354/

tem with optimality or stationarity guarantees). These algorithms are rooted in the operator splitting theory of optimization (Lions and Mercier, 1979; Eckstein and Bertsekas, 1992), which seeks to solve monolithically difficult problems through decomposition into easier subproblems. For example, the classical alternating direction method of multipliers (ADMM) algorithm (Boyd et al., 2011), when used in distributed MPC, uses an augmented Lagrangean formulation to handle the interactions among subsystems; the subsystem controller decisions and Lagrange multipliers are iteratively updated based on the augmented Lagrangean (Mota et al., 2014; Farokhi et al., 2014). As such, the control performance of MPC, which relies on the quality of the solution found (Mayne et al., 2000), is guaranteed provided that the distributed optimization iterations converge to a monolithic optimum.

Computational time is an important consideration in the practical implementation of MPC. Specifically, the allowed computational time for MPC must be small enough compared to the sampling interval. For centralized MPC, extensive works have focused on developing efficient and structure-exploiting algorithms (Biegler and Zavala, 2009; Patterson and Rao, 2014) as well as modified variants of MPC that avoid full solution routines at all sampling times, e.g., by exploiting the parametric sensitivity information or by carrying out truncated iterations in the centralized nonlinear programming solver (Lopez-Negrete et al., 2013; Wolf and Marquardt, 2016). For distributed MPC based on distributed optimization, fully completing the computation within the allowed time becomes even more challenging, since typically tens to hundreds of iterations of subsystem optimization are needed. Some approaches to address this challenge include:

- Using accelerated distributed optimization algorithms, of which the momentum method and Anderson acceleration are representative (Pu et al., 2014; Wang and Ong, 2018; Tang and Daoutidis, 2019b; 2021b).
- Fine-tuning distributed optimization with metric selection methods to reduce the required computational effort (Ghadimi et al., 2014; Giselsson and Boyd, 2016).
- Using distributed versions of fast centralized MPC methods, e.g., by putting difficult steps such as evaluating Hessian and identifying active constraints offline and leaving only easy steps online (Cai et al., 2014; Yu et al., 2019).
- Developing algorithms that need fewer iterations, e.g., the AL-ADIN algorithm (Houska et al., 2016) uses a semi-centralized scheme that collects sensitivity information from subsystems for coordination based on quadratic programming (QP).
- Allowing solutions with inexactness that is small enough to guarantee recursive feasibility and continually tightened for asymptotic stability (called constraint tightening) (Giselsson and Rantzer, 2013; Rubagotti et al., 2014).

In this paper, we adopt the perspective of *early termination*, i.e., we consider a distributed optimization algorithm for MPC that can be terminated after any number of iterations before fully converging to an optimum. The goal is to meet the computational time restrictions, yet still returning control actions that guarantee closed-loop stability and improve over the ones otherwise obtained under any earlier termination. Early termination requires a primal instead of primal-dual algorithm to handle the inter-subsystem constraints. Such an idea has been used in stochastic MPC under non-anticipativity constraints (Krishnamoorthy et al., 2019). For distributed optimization in MPC, we propose to use a *splitting algorithm*, analogous to forward-backward splitting (FBS) (Lions and Mercier, 1979; Chen and Rockafellar, 1997; Duchi and Singer, 2009) but with the forward gradient-descent step replaced by a proximal step with ℓ_1 and quadratic penalty terms. Specifically, the optimal control problem in MPC is formulated as an optimization problem whose decision variables are the shared variables, i.e., the predicted trajectories of the interconnecting signals between subsystems. Each iteration includes a distributed controller solution step, which finds the subsystems' proximal solution to the shared variables, and a coordination step to update the shared variables. The iterations in the splitting algorithm are allowed to be early terminated according to computational time restrictions.

Key to the performance of such a splitting algorithm is the concept of *Lyapunov envelope*. The essential idea is that when distributed optimization iterations have not yet converged and hence the inter-subsystem equality constraints are violated, an estimate of the effect of such inter-subsystem violations on the plantwide system in terms of an upper bound of the system's control-Lyapunov function can still be obtained. Here we derive this estimate based on the assumption that the subsystems are incrementally dissipative in their interconnecting inputs/outputs from/to other subsystems, which implies a finite gain of inter-subsystem violations, considered as disturbances, on the objective function of the plantwide MPC problem. Thus, if the control actions are executed, the resulting predicted trajectory, namely the "consolidated solution" (Ferranti and Keviczky, 2015) will have a corresponding change from the subsystems' predictions with upper-bounded increase in the control-Lyapunov function. Therefore, by constructing the splitting algorithm such that the proximal formulation (consisting of the original function and the violations to ℓ_1 and quadratic penalties of interconnecting equalities) is consistent with this overestimate of the plant-wide control-Lyapunov function, which we call the Lyapunov envelope, the iterations will result in monotonic decrease of the Lyapunov envelope and the early termination will not deteriorate closed-loop stability.

Compared to other techniques, the most distinctive feature of the Lyapunov envelope algorithm is its flexibility – it can be implemented with different extents of restrictions on computational time or number of iterations, thus enabling the user to trade-off the computational and control performance of distributed MPC. Such a characteristic also tolerates inexactness of subsystem solvers, and does not have specific requirements on the subsystem solver. The iterations involve only minimal communication, i.e., no information exchange other than the primal variable values pertaining to the interactions among subsystems is needed. Thus, the proposed Lyapunov envelope algorithm is a promising approach to the practical real-time implementation of distributed MPC.

The paper is organized as follows. Preliminaries and MPC formulations are first introduced in Section 2, where we will review the basic assumptions for the closed-loop stability of distributed MPC. Then FBS as a classical distributed optimization algorithm is introduced in Section 3 and we discuss its potential and limitations in addressing real-time implementation of distributed MPC. Subsequently, through the analysis of the consolidated solution and the system dissipativity, the concept of Lyapunov envelope is derived and the Lyapunov envelope algorithm is presented in Section 4. A case study on a vinyl acetate monomer process is considered in Section 5, which, to our best knowledge, is *the first investigation of distributed MPC on such an industrial-scale chemical plant*. Finally, conclusions are given in Section 6.

Notations. We use $\mathbb{I}_n$ to denote the set of nonnegative integers up to n: $\{0, 1, \ldots, n\}$ and $\mathbb{I}_n^+ = \mathbb{I}_n \backslash \{0\}$. Subscripts $i = 1, \ldots, n$ are used for subsystems, superscripts $r = 0, 1, \ldots$ are used for iterations of optimization algorithm, and the bracketed index $[t]$ is the time stamp. A $\mathcal{K}_\infty$-class function $\alpha : [0, \infty) \to [0, \infty)$ refers to a strictly increasing continuous function satisfying $\alpha(0) = 0$ and $\lim_{r\to\infty} \alpha(r) = \infty$. An indicator function $\mathbf{1}(\cdot)$ takes a value of 0 if the proposition in the parentheses holds true and ∞ otherwise. We also use $\mathbf{1}_\Sigma(s) = \mathbf{1}(s \in \Sigma)$ for any set Σ. Blackboard bold Latin letters and capitalized Greek letters stand for sets. Scalars, vectors, and functions are represented using lower case letters, and matrices use capitalized Latin letters. We denote by $v = [v_1; v_2; \ldots; v_n]$

the vertical stacking of column vectors $v_1, v_2, \ldots, v_n$ into a vector v. A matrix $M \succeq (\preceq) 0$ stands for positive (negative) semidefiniteness, and $M \succ (\prec) 0$ stands for positive (negative) definiteness. When $M \succ 0$, the quadratic form $x^\top Mx$ is represented by $\|x\|_M^2$, and specifically when $M = I$, $\|x\|_M^2$ is simply written as $\|x\|^2$ The minimum and maximum eigenvalues of M are denoted by $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$, respectively. Clearly, $\lambda_{\min}(M)\|x\|^2 \le \|x\|_M^2 \le \lambda_{\max}(M)\|x\|^2$.

2. Preliminaries and formulations

2.1. Centralized MPC

We consider a discrete-time system P comprising of n subsystems. For subsystem i, $i = 1, 2, \ldots, n$, the dynamics is written as

$$P_i : \begin{cases} x_i[t+1] & = f_i(x_i[t], u_i[t], w_i[t]) \\ v_i[t] & = h_i(x_i[t], u_i[t]) \end{cases} \tag{1}$$

where x_i, u_i, w_i, and v_i stands for the vector of states, inputs, interconnecting inputs from other subsystems, and interconnecting outputs acting on other subsystems, respectively; the functions f_i and h_i are continuous. In other words, P_i maps signals w_i and u_i into v_i and x_i. We assume that all the state values are available for control. Suppose that the interconnections among the subsystems are specified by the linear relations with matrices H_{ij}:

$$w_i = H_{ij} v_j, \ i, j = 1, \ldots, n, \ i \neq j. \tag{2}$$

By vertically stacking the vectors of the subsystems, the dynamics of the entire system, comprising of the disconnected subsystem dynamics $P = (P_1, \ldots, P_n)$ that maps $(w, u) \to (v, x)$ and the interconnections $H : v \to w$, can be denoted by $P_H : u \to x$ and expressed as

$$P_H : \begin{cases} x[t+1] & = f(x[t], u[t], w[t]) \\ v[t] & = h(x[t], u[t]) \\ w[t] & = Hv[t] \end{cases} \tag{3}$$

or simply

$$x[t+1] = f(x[t], u[t], Hh(x[t], u[t])) =: f_H(x[t], u[t]). \tag{4}$$

In centralized MPC, at time k, the control action $u[k]$ is determined by solving monolithically the following optimization problem that minimizes the cost associated with the predicted trajectory of x and u, denoted as $\hat{x}$ and $\hat{u}$, respectively, and executing $u[k] = \hat{u}[0]$:

$$\begin{array}{ll} \min & \sum_{t=0}^{N-1} \ell(\hat{x}[t], \hat{u}[t]) + \ell_{\mathrm{f}}(\hat{x}[N]) \\ \text{s.t.} & \hat{x}[t+1] = f(\hat{x}[t], \hat{u}[t], \hat{w}[t]), \ t \in \mathbb{I}_{N-1} \\ & \hat{v}[t] = h(\hat{x}[t], \hat{u}[t]), \ t \in \mathbb{I}_{N-1} \\ & \hat{w}[t] = H\hat{v}[t], \ t \in \mathbb{I}_{N-1} \\ & \hat{x}[t] \in \mathbb{X}, \hat{u}[t] \in \mathbb{U}, \ t \in \mathbb{I}_{N-1} \\ & \hat{x}[0] = x[k], \ \hat{x}[N] \in \mathbb{X}_{\mathrm{f}}. \end{array} \tag{5}$$

In the above formulation, $\mathbb{X}$, $\mathbb{U}$, $\mathbb{X}_{\mathrm{f}}$ are the state, input and terminal constraints, respectively; ℓ and ℓ_{f} are the stage cost and terminal cost functions, respectively; N is the prediction horizon length. Let us recall the classical result for the closed-loop asymptotic stability of centralized MPC (see, e.g., Section 3 of Mayne et al. (2000)).

Theorem 1. *Suppose that*

1. *$f(0,0) = 0$, $h(0,0) = 0$, $\ell(0,0) = 0$, $\ell_{\mathrm{f}}(0) = 0$, $\mathbb{X}$ is closed, and $\mathbb{U}$ and $\mathbb{X}_{\mathrm{f}} \subseteq \mathbb{X}$ are compact, containing the origin in their interiors;*
2. *there exist $\mathcal{K}_\infty$-class functions α and α_{f} such that $\ell(x,u) \ge \alpha(\|x\|), \forall x \in \mathbb{X}, \forall u \in \mathbb{U}$ and $\ell_{\mathrm{f}} \le \alpha_{\mathrm{f}}(\|x\|), \forall x \in \mathbb{X}_{\mathrm{f}}$;*
3. *there exists a control law κ_{f}, such that $\forall x \in \mathbb{X}_{\mathrm{f}}$, $\kappa_{\mathrm{f}}(x) \in \mathbb{U}$, $f_H(x, \kappa_{\mathrm{f}}(x)) \in \mathbb{X}_{\mathrm{f}}$, and $\ell_{\mathrm{f}}(x, f_H(x, \kappa_{\mathrm{f}}(x))) - \ell_{\mathrm{f}}(x) \le -\ell(x, \kappa_{\mathrm{f}}(x))$.*

Then by solving the centralized MPC problem (5) parameterized by $x[k]$ and denoting the resulting control law specified by $u[k] = \hat{u}[0]$ as $u = \kappa(x)$, we have the following statements hold true:

1. *(Recursive feasibility.) If the problem (5) is feasible for $x[t]$, then for $x[t+1] = f_H(x[t], \kappa(x[t]))$ it remains feasible.*
2. *(Lyapunov descent.) The optimized objective function $V(x[t])$ satisfies $V(x[t+1]) - V(x[t]) \le -\ell(x[t], \kappa(x[t]))$.*

It then follows that V is a control-Lyapunov function yielding asymptotic closed-loop stability, i.e., starting from any state $x[k]$ that makes the problem (5) feasible, the state trajectory converges to the origin.

2.2. Formulation of distributed MPC

In the context of distributed MPC where we need to optimize control actions on the basis of subsystems, we first make the following separability assumption on the subsystems for simplicity, which allows the reformulation of problem (5) without involving constraints across subsystems other than the interconnections $w = Hv$.

Assumption 1. Suppose that the following statements hold:

1. In (5), ℓ, ℓ_{f}, $\mathbb{X}$, $\mathbb{X}_{\mathrm{f}}$, and $\mathbb{U}$ can be separated into subsystems, i.e., $\ell(x,u) = \sum_{i=1}^n \ell_i(x_i, u_i)$, $\ell_{\mathrm{f}}(x) = \sum_{i=1}^n \ell_{\mathrm{fi}}(x_i)$, $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$, $\mathbb{X}_{\mathrm{f}} = \mathbb{X}_{\mathrm{f}1} \times \cdots \times \mathbb{X}_{\mathrm{f}n}$, $\mathbb{U} = \mathbb{U}_1 \times \cdots \times \mathbb{U}_n$.
2. For each $i = 1, \ldots, n$, $f_i(0,0) = 0$, $h_i(0,0) = 0$, $\ell_i(0,0) = 0$, $\ell_{\mathrm{fi}}(0) = 0$, $\mathbb{X}_i$ is closed, and $\mathbb{U}_i$ and $\mathbb{X}_{\mathrm{fi}} \subseteq \mathbb{X}_i$ are compact, containing the origin in their interiors;
3. For each $i = 1, \ldots, n$, there exist $\mathcal{K}_\infty$-class functions α_i and α_{fi} such that $\ell_i(x_i, u_i) \ge \alpha_i(\|x_i\|), \forall x_i \in \mathbb{X}_i, \forall u_i \in \mathbb{U}_i$ and $\ell_{\mathrm{fi}} \le \alpha_{\mathrm{fi}}(\|x_i\|), \forall x_i \in \mathbb{X}_{\mathrm{fi}}$.

Hence the MPC problem becomes

$$\begin{array}{ll} \min & \sum_{i=1}^n \left(\sum_{t=0}^{N-1} \ell_i(\hat{x}_i[t], \hat{u}_i[t]) + \ell_{\mathrm{fi}}(\hat{x}_i[N]) \right) \\ \text{s.t.} & \hat{x}_i[t+1] = f_i(\hat{x}_i[t], \hat{u}_i[t], \hat{w}_i[t]), \ t \in \mathbb{I}_{N-1}, i \in \mathbb{I}_n^+ \\ & \hat{v}_i[t] = h_i(\hat{x}_i[t], \hat{u}_i[t]), \ t \in \mathbb{I}_{N-1}, i \in \mathbb{I}_n^+ \\ & \hat{w}[t] = H\hat{v}[t], \ t \in \mathbb{I}_{N-1} \\ & \hat{x}_i[t] \in \mathbb{X}_i, \hat{u}_i[t] \in \mathbb{U}_i, \ t \in \mathbb{I}_{N-1}, i \in \mathbb{I}_n^+ \\ & \hat{x}_i[0] = x_i[k], \ \hat{x}_i[N] \in \mathbb{X}_{\mathrm{fi}}, i \in \mathbb{I}_n^+. \end{array} \tag{6}$$

In the above formulation, $\hat{w}[t] = H\hat{v}[t]$ is not separable into subsystems. Hence we create *shared variables* $s = [s_1; \ldots; s_n] = [s_{w1}; s_{v1}; \ldots; s_{wn}; s_{vn}]$, where s_{wi} is a copy of $[\hat{w}_i[0]; \ldots; \hat{w}_i[N-1]]$ and analogously s_{vi} is a copy of $[\hat{v}_i[0]; \ldots; \hat{v}_i[N-1]]$. The inseparable constraints are hence rewritten as constraints between subsystems and shared variables $\hat{w}_i[t] = s_{wi}[t]$, $\hat{v}_i[t] = s_{vi}[t]$, and a constraint on the shared variables $s_{wi}[t] = \sum_{j=1}^n H_{ij} s_{vi}[t]$. More compactly, by stacking all variables in each subsystem i into a vector ξ_i and the subsystem constraints as Ξ_i (depending on $x_i[t]$), rewriting the linear equality constraints between the subsystems and the shared variables as $C_i\xi_i = s_i$ (for some properly constructed matrices C_i, $i = 1, 2, \ldots, n$), expressing the constraints on s as $s \in \Sigma$ (which is the kernel of some matrix), and the objective function of subsystem i as ϕ_i, we arrive at the following formulation:

$$\begin{array}{ll} \min_{\xi, s} & \sum_{i=1}^n \phi_i(\xi_i) \\ \text{s.t.} & \xi_i \in \Xi_i, C_i\xi_i = s_i, \ i \in \mathbb{I}_n^+ \\ & s = [s_1; \ldots; s_n] \in \Sigma. \end{array} \tag{7}$$

This formulation has a structure as illustrated by Fig. 1, which we refer to as the *agent-coordinator architecture*.

Specifically, the decision variables of the subsystems ξ_i subject to constraints Ξ_i (which may be complex) are handled separately and in parallel by their own optimization solvers or routines. These subsystem solvers are called *agents*, which should be capable of

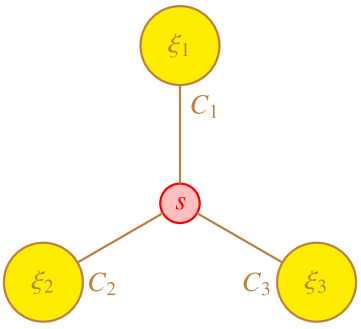

Fig. 1. Agent-coordinator architecture.

solving quadratic programming or nonlinear programming problems depending on linear or nonlinear system dynamics in (1). The shared variables s which account for the interactions across subsystems through linear equality constraints, are handled by another solver, called *coordinator*. Compared to the agents, the coordinator is only required to use a simple procedure to update the shared variables after collecting some information from the subsystem solutions.

If we further denote the optimized objective function value of the subsystems' optimization problem ($+\infty$ if infeasible)

$$\begin{array}{ll}\min_{\xi_i} & \phi_i(\xi_i) \\ \text{s.t.} & \xi_i \in \Xi_i,\ C_i\xi_i = s_i\end{array} \tag{8}$$

as $\varphi_i(s_i)$ for $i = 1, \ldots, n$, then the distributed MPC problem (7) can be simply written as a constrained optimization problem of the shared variables s:

$$\min_{s\in\Sigma} \sum_{i=1}^{n} \varphi_i(s_i). \tag{9}$$

Hence to solve (7), an iterative algorithm is adopted to update the value of s, and in each iteration, the associated values of the decision variables of the subsystems that solve the optimization problems (8) are found. Before discussing the algorithm, we consider how any algorithm will affect the closed-loop stability.

2.3. Stability of distributed MPC

Key to the establishment of the Lyapunov descent property in Theorem 1 is the observation that if at time t, the MPC predicts a control sequence $\hat{u}[0], \ldots, \hat{u}[N-1]$ with predicted states $\hat{x}[1], \ldots, \hat{x}[N]$, then at time $t+1$, the MPC can be initialized at a feasible solution by doing the following replacement:

$$\begin{array}{l}\hat{u}[0], \hat{u}[1], \ldots, \hat{u}[N-1] \leftarrow \hat{u}[1], \ldots, \hat{u}[N-1], \kappa_f(\hat{x}[N]), \\ \hat{x}[1], \hat{x}[2], \ldots, \hat{x}[N] \leftarrow \hat{x}[2], \ldots, \hat{x}[N], f_H(\hat{x}[N], \kappa_f(\hat{x}[N])),\end{array} \tag{10}$$

i.e., by extending the predicted trajectory for one sampling time under the control law κ_f. This initialization already achieves the descent property with an objective function descent of at least $-\ell(x[t], u[t])$, which can only be further improved with optimization. For distributed optimization, we still assume the existence of such an a-priori basic stabilizing control law κ_f.

Assumption 2. There exists a control law κ_f, such that $\forall x \in \mathbb{X}_f$, $\kappa_f(x) \in \mathbb{U}$, $f_H(x, \kappa_f(x)) \in \mathbb{X}_f$, and $\ell_f(x) - \ell_f(x, f_H(x, \kappa_f(x))) \geq \ell(x, \kappa_f(x))$.

Then by initializing the distributed optimization with the same extension rule (10), the condition under which the closed-loop stability is not compromised is the following.

Lemma 1 (Closed-loop stability under distributed optimization)**.** *Suppose that Assumptions 1 and 2 hold. If for any $x[t]$ such that problem (7) is feasible with any feasible initialization of $\xi_1, \ldots, \xi_n, s$, the distributed optimization algorithm returns a feasible solution of (7) with decreased objective function from the initialization using the extension under κ_f, then under the distributed optimization algorithm, distributed MPC keeps the Lyapunov descent and asymptotic stability properties.*

In the above lemma, we need the distributed optimization algorithm to return a feasible solution, since the feasible solution is *consolidated*, i.e., implementing the predicted control sequence will steer the system onto the predicted states, thus allowing the extension (10) that makes V a Lyapunov function. In other words, we require the distributed optimization algorithm to achieve, by solving (9), a descent on the following Lyapunov function, defined as the summation of the subsystems' objective function values and an indicator function of the interconnecting constraints:

$$V(x[t]) = \sum_{i=1}^{n} \left(\phi_i(\xi_i) + \mathbf{1}_{\Xi_i}(\xi_i) + \mathbf{1}(C_i\xi_i = s_i)\right) + \mathbf{1}_{\Sigma}(s). \tag{11}$$

Letting $\psi(s) = \mathbf{1}_{\Sigma}(s)$, (9) can be equivalently expressed as

$$\min_{s}\ \varphi(s) + \psi(s) = \sum_{i=1}^{n} \varphi_i(s_i) + \psi(s). \tag{12}$$

The formulation (12) lends itself to an agent-coordinator architecture, as the functions φ_i are defined based on the subsystem problems (8) and $\psi = \mathbf{1}_{\Sigma}$ is a function associated with the interconnections among the subsystems. Next we introduce the forward-backward splitting (FBS) algorithm for solving (12).

3. Forward-backward splitting (FBS)

3.1. The original FBS algorithm

For minimizing a composite function $\varphi + \psi$ as in (12), the FBS algorithm iterates the s variables by

$$\begin{array}{ll}s^{r+1/2} & = s^r - \gamma\nabla\varphi(s^r), \\ s^{r+1} & \in \text{prox}_{\gamma\psi}(s^{r+1/2}),\ r = 0, 1, \ldots\end{array} \tag{13}$$

where $\gamma > 0$ is the step size, and the proximal operator for any $\gamma > 0$ and function g is defined by

$$\text{prox}_{\gamma g}(s) = \arg\min_{s'} \left(g(s') + \frac{1}{2\gamma}\|s' - s\|^2\right). \tag{14}$$

Particularly for ψ as an indicator function of Σ, the proximal operator is independent of γ and is a projection:

$$\text{prox}_{\gamma\psi}(s) = \text{proj}_{\Sigma}(s) = \arg\min_{s'\in\Sigma} \|s' - s\|^2. \tag{15}$$

In other words, each iteration contains a gradient descent step and a projection step. The FBS has a sufficient descent property as follows (Bolte et al., 2014, Proposition 2). With increasing iterations, this descent property implies the convergence of s^r towards a stationary point (Themelis et al., 2018), which is not detailed here since closed-loop stability of distributed MPC does not necessarily need the solution convergence.

Lemma 2 (Descent property of FBS)**.** *For any function φ that is continuously differentiable with a Lipschitz continuous gradient $\nabla\varphi$ (denote the Lipschitz constant as l) and any proper, closed, lower-bounded function ψ, when $\gamma < 1/l$, under the FBS iteration (13) we have*

$$\varphi(s^{r+1}) - \varphi(s^r) \leq -\frac{1 - \gamma l}{2\gamma}\|s^{r+1} - s^r\|^2. \tag{16}$$

We note that the FBS algorithm may not directly satisfy the condition of the above lemma, since the functions φ_i may not be

differentiable and when s_i is such that (8) is infeasible, $\varphi_i(s_i) = \infty$. To show this complexity, we denote by Σ_i the region of s_i such that the problem for subsystem i (8) has a feasible solution, i.e.,

$$\Sigma_i = \{C_i\xi_i | \xi_i \in \Xi_i\},\ i \in \mathbb{I}_n^+, \tag{17}$$

and

$$\Sigma' = \Sigma \cap (\Sigma_1 \cap \cdots \cap \Sigma_n). \tag{18}$$

Then for any $s \in \Sigma'$, all $\phi_i(s_i)$ have a finite value. It suffices to iterate s in Σ'. That is, instead of solving (12) with $\psi = \mathbf{1}_\Sigma$, we should redefine ψ as $\mathbf{1}_{\Sigma'}$ and carry out FBS as

$$s^{r+1} \in \text{proj}_{\Sigma'}(s^r - \gamma\nabla\varphi(s^r)),\ r = 0, 1, \ldots. \tag{19}$$

This in fact demands the following conditions for using the FBS algorithm.

Assumption 3. Suppose that

1. the functions $\nabla\varphi(s_i)$, namely the gradients of the optimized objective function values of (8) depending on s_i as parameters, exist and can be computed for $i = 1, \ldots, n$, and that
2. the projection operation onto Σ' as the intersection of $n+1$ sets can be computed without an alternating projection procedure to evaluate proj_{Σ_i}, $i = 1, \ldots, n$ iteratively.

Corollary 1 (Closed-loop stability under FBS algorithm)**.** *Under Assumption 3, the FBS iteration (19) satisfies the sufficient descent property (16), which implies closed-loop stability of distributed MPC under Assumptions 1 and 2 on the system (1).*

However, the conditions of Assumption 3 are difficult to satisfy. First, the evaluation of $\nabla\varphi_i$ needs a parametric sensitivity analysis of the subsystem problems (8), which rely on further regularity assumptions and solvers that evaluate such sensitivities (Fiacco and Ishizuka, 1990, Section 5) or an identification scheme of active constraints in the subsystems' problems (Quirynen and Di Cairano, 2020). Second, the finite-time exact evaluation of $\text{proj}_{\Sigma'}$ by the coordinator requires an off-line explicit characterization of Σ', which is possible only in simple situations (e.g., when each subsystem is controllable, single-input-single-output, and unconstrained in u_i) and usually can not be easily obtained.

3.2. Modified FBS algorithm

The difficulties in satisfying Assumption 3 fundamentally originate from the interconnecting constraints $C_i\xi_i = s_i$, $i \in \mathbb{I}_n^+$, linking the agents and the coordinator. In the FBS algorithm introduced in the previous subsection, the interconnecting constraints are required to be satisfied *exactly* throughout the iterations so that the descent of $\varphi(s)$ yields a Lyapunov descent. Therefore, it is desirable to relax these constraints to obtain *inexact* solutions. The idea is natural in that even when $C_i\xi_i - s_i$, $i = 1, \ldots, n$, are not exactly zero, as long as they are sufficiently small in magnitude, the solution is still approximately feasible and may be tolerable.

In other words, we seek to replace the $\mathbf{1}(C_i\xi_i = s_i)$ terms in the Lyapunov function (11) with a "soft constraint" represented by a penalty term $\varpi(C_i\xi_i - s_i)$, where ϖ is a positive definite and even function satisfying $\varpi(0) = 0$. Denote $C = \text{diag}(C_1, \ldots, C_n)$ and $\varpi(C\xi - s) = \sum_{i=1}^n \varpi(C_i\xi_i - s_i)$. With such a penalty ϖ, we propose to substitute the gradient descent step in (13) with the following proximal step:

$$s^{r+1/2} \in \text{prox}_\varphi^\varpi(s^r) := \arg\min_s (\varphi(s) + \varpi(s - s^r)). \tag{20}$$

Given any s^r, the solution of (20) is hence carried out by agents ($i \in \mathbb{I}_n^+$) in parallel without needing to evaluate gradients of φ_i:

$$\begin{aligned} \xi_i^r \in\ & \arg\min_{\xi_i}\ \phi_i(\xi_i) + \varpi(C_i\xi_i - s_i^r) \\ & \text{s.t. } \xi_i \in \Xi_i \\ s_i^{r+1/2} =\ & C_i\xi_i^r. \end{aligned} \tag{21}$$

The second line in (13) is also modified with ϖ:

$$s^{r+1} \in \text{prox}_\psi^\varpi(s^{r+1/2}) = \arg\min_{s\in\Sigma} \varpi(s - s^{r+1/2}). \tag{22}$$

which can be executed by the coordinator without knowing the monolithic feasible set Σ' (18).

The modified FBS algorithm, comprising of two proximal steps (21) and (22) in each iteration, is not an algorithm that guarantees convergence to the optimum with an increasing number of iterations. Nevertheless, since the two steps guarantee $\varphi(s^{r+1/2}) + \varpi(s^{r+1/2} - s^r) \le \varphi(s^{r-1/2}) + \varpi(s^{r-1/2} - s^r)$ and $\varpi(s^{r+1} - s^{r+1/2}) \le \varpi(s^r - s^{r+1/2})$, respectively for all $r \ge 1$, we have $\varphi(s^{r+1/2}) + \varpi(s^{r+1} - s^{r+1/2}) \le \varphi(s^{r-1/2}) + \varpi(s^r - s^{r-1/2})$. Now, to remove fractional superscripts, let us denote s^{r+1} by $\hat{s}^{r+1}$ and $s^{r+1/2}$ by $\bar{s}^{r+1}$ ($r = 0, 1, \ldots$) to represent the shared variables s solved by the agents and the coordinator respectively. The modified FBS thus guarantees that the iterations result in a monotonic decrease in the value of $\varphi(\hat{s}) + \varpi(\bar{s} - \hat{s})$.

Yet, the question lies in: *how should the penalty function ϖ be chosen, so that the descent of $\phi(\hat{s}) + \varpi(\bar{s} - \hat{s})$ is still a Lyapunov descent?* This is the key issue to be answered by the Lyapunov envelope approach in the subsequent section.

4. Lyapunov envelope

4.1. Analysis of consolidated solution

Following the notations from the previous section, suppose that under the modified FBS algorithm ((21) and (22)) within a certain number of iterations, we have obtained from the agents a solution of $(\xi_1, \ldots, \xi_n, \hat{s})$, satisfying $\xi_i \in \Xi_i$ and $C_i\xi_i = \hat{s}_i$ for all $i \in \mathbb{I}_n^+$, and from the coordinator an $s = \bar{s} \in \Sigma$ that does not equal $\hat{s}$ exactly. In other words, the algorithm returns a solution (ξ, s) that violates the interconnection constraints $C\xi - s = 0$ ($\hat{s} - \bar{s} = 0$), i.e., is not consolidated. Therefore, $\varphi(\hat{s}) = \sum_{i=1}^n \phi(\xi_i)$ is no longer the cost of an actual future trajectory and a control-Lyapunov function. To obtain a control-Lyapunov function in the form of $\varphi(\hat{s}) + \varpi(\bar{s} - \hat{s})$, we need to analyze the effect of nonzero $C\xi - s = \hat{s} - \bar{s}$ (violations to the interconnecting constraints) on the control-Lyapunov function of the consolidated solution.

Now we denote

$$\begin{aligned} \hat{s}_i &= \left[\hat{w}_i[k]; \ldots; \hat{w}_i[k+N-1]; \hat{v}_i[k]; \ldots; \hat{v}_i[k+N-1]\right], \\ \bar{s}_i &= [\bar{w}_i[k]; \ldots; \bar{w}_i[k+N-1]; \bar{v}_i[k]; \ldots; \bar{v}_i[k+N-1]] \end{aligned} \tag{23}$$

The former vector stands for the values of the interconnecting variables solved by the subsystems' agents, which conform with the subsystem model, while the latter vector stands for the values of interconnecting variables solved by the agent, which conform with the interconnecting relations H. Their differences are denoted by vectors δ_w and δ_v comprising of the following entries:

$$\delta_{wi}[t] = \hat{w}_i[t] - \bar{w}_i[t],\ \delta_{vi}[t] = \hat{v}_i[t] - \bar{v}_i[t],\ t \in \mathbb{I}_{N-1}, i \in \mathbb{I}_n^+. \tag{24}$$

The solution obtained by the optimization algorithm is hence a prediction of the system as shown by the inner block diagram of Fig. 2. When δ_w and δ_v are nonzero, the predicted trajectory is realized on a hypothetical system interconnected by Σ and H with additional signals δ_w and $-\delta_v$ imposed before and after P to account for the discrepancies between the agents and the coordinator.

The consolidated solution, which is the trajectory of the actual system without the hypothetical signals, is hence the predicted solution with opposite hypothetical signals $-\delta_w$ and δ_v imposed to set off δ_w and $-\delta_v$, respectively, as shown by the red area in Fig. 2. Therefore, the consolidated solution can be regarded as a perturbation of the predicted trajectory with exogenous disturbances. It is hence desirable to analyze the effect of such disturbances on the objective function of the solution returned from distributed optimization.

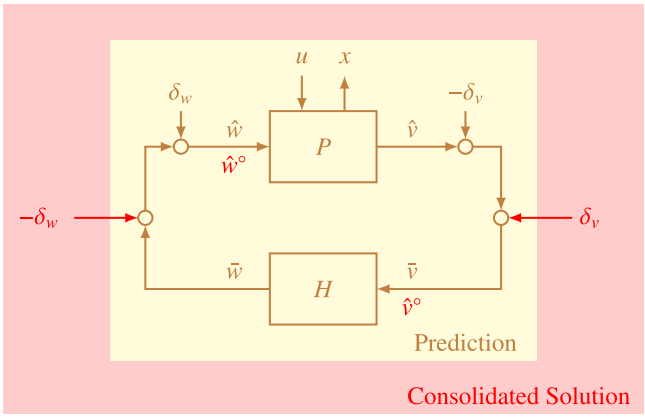

Fig. 2. Consolidated solution viewed as a disturbed system.

4.2. Analysis of system dissipativity

Dissipativity provides a natural framework for analyzing the stability of interconnected systems (Brogliato et al., 2020). Specifically, for the analysis of the perturbed system, we adopt the concept of incremental dissipativity (Pavlov and Marconi, 2008) here.

Assumption 4. For $i \in \mathbb{I}_n$, subsystem P_i is incrementally dissipative. In other words, for any $u_i[t] \in \mathbb{U}_i$, and any two different $w_i[t]$ and $x_i[t] \in \mathbb{X}_i$, there exist matrices $M_i, \Omega_{wi}, \Omega_{vi} \succeq 0$, such that the differences between the two $w_i[t]$, $x_i[t]$ and the resulting two $x_i[t+1]$ and $v_i[t]$, denoted as $\delta_{wi}[t]$, $\delta_{xi}[t]$, $\delta_{xi}[t+1]$ and $\delta_{vi}[t]$, respectively, satisfy

$$\|\delta_{xi}[t+1]\|^2_{M_i} - \|\delta_{xi}[t]\|^2_{M_i} \le \|\delta_{wi}[t]\|^2_{\Omega_{wi}} - \|\delta_{vi}[t]\|^2_{\Omega_{vi}}. \tag{25}$$

For example, suppose that P_i is a linear system:

$$P_i : \begin{cases} x_i[t+1] &= A_i x_i[t] + B_i u_i[t] + E_i w_i[t] \\ v_i[t] &= C_i x_i[t] + D_i u_i[t] + F_i w_i[t]. \end{cases} \tag{26}$$

Then its increment can be written as

$$\delta P_i : \begin{cases} \delta_{xi}[t+1] &= A_i \delta_{xi}[t] + E_i \delta_{wi}[t] \\ \delta_{vi}[t] &= C_i \delta_{xi}[t] + F_i \delta_{wi}[t], \end{cases} \tag{27}$$

which satisfies the incrementally dissipative inequality (25) if the following linear matrix inequality (LMI) is satisfied for M_i, Ω_{wi}, $\Omega_{vi} \succeq 0$:

$$\begin{bmatrix} A_i^\top M_i A_i - M_i + C_i^\top \Omega_{vi} C_i & A_i^\top M_i E_i + C_i^\top \Omega_{vi} F_i \\ E_i^\top M_i A_i^\top + F_i^\top \Omega_{vi} C_i & E_i^\top M_i E_i - \Omega_{wi} + F_i^\top \Omega_{vi} F_i \end{bmatrix} \preceq 0. \tag{28}$$

For the stage cost and terminal cost functions in the objective function of the MPC problem (5), ℓ and ℓ_f, we assume the following conditions.

Assumption 5. For all $i \in \mathbb{I}^+_n$,

1. functions ℓ_i and ℓ_{fi} are continuously differentiable in the x_i argument;
2. the gradients satisfy $\partial \ell(0, u_i)/\partial x_i = 0$ and for any $u_i \in \mathbb{U}_i$, $\partial \ell_{fi}(0, u_i)/\partial x_i = 0$;
3. the gradients of ℓ_i and ℓ_{fi} are Lipschitz with constants l_i and l_{fi}, respectively, i.e.,
$$\begin{aligned} \left\| \tfrac{\partial \ell_i}{\partial x_i}(\hat{x}_i, \hat{u}_i) - \tfrac{\partial \ell_i}{\partial x_i}(\hat{x}_i^\circ, \hat{u}_i) \right\| &\le l_i \|\hat{x}_i^\circ - \hat{x}_i\|, \\ \left\| \tfrac{d \ell_{fi}}{d x_i}(\hat{x}_i) - \tfrac{d \ell_{fi}}{d x_i}(\hat{x}_i^\circ) \right\| &\le l_{fi} \|\hat{x}_i^\circ - \hat{x}_i\|, \end{aligned} \tag{29}$$
for any $\hat{x}_i, \hat{x}_i^\circ$ and $\hat{u}_i \in \mathbb{U}_i$;
4. the $\mathcal{K}_\infty$-class function α_i lower bounding ℓ_i can be found as a squared norm: $\alpha_i(x_i) = \frac{g_i}{2}\|x_i\|^2$ for some constant $g_i > 0$, and ℓ_{fi} is also lower bounded by $\frac{g_{fi}}{2}\|x_i\|^2$ for some constant $g_{fi} > 0$.

It is straightforward to verify that the above assumption is satisfied by a typical choice of quadratic cost functions $\ell_i(x_i, u_i) = \|x_i\|^2_{Q_i} + \|u_i\|^2_{R_i}$, $\ell_{fi}(x) = \|x_i\|^2_{S_i}$ with $Q_i, R_i \succ 0$, $S_i \succeq 0$.

Now let us consider the effect of disturbances (δ_w, δ_v) on the states of the consolidated solution. We denote the interconnecting variables in the consolidated solution by $\hat{w}^\circ[t]$ and $\hat{v}^\circ[t]$, and let $\delta^\circ_{wi}[t] := \hat{w}^\circ_i[t] - \hat{w}_i[t]$, $\delta^\circ_{vi}[t] := \hat{v}^\circ_i[t] - \hat{v}_i[t]$ be the differences between the consolidated solution and the predicted solution, stacked into δ°_w and δ°_v, respectively. We stack the matrices involved in subsystems' incremental dissipativity properties into block diagonal forms $\Omega_w = \mathrm{diag}(\Omega_{w1}, \ldots, \Omega_{wn})$ and analogously Ω_v and M. Then we have the following bound.

Lemma 3 (Effect of violations to the interconnecting constraints on the states of the consolidated solution)**.** *Suppose that Assumption 4 holds and in addition*

$$\Omega_v - 2H^\top \Omega_w H \succ 0. \tag{30}$$

Then for any $T \in \mathbb{I}_{N-1}$, we have

$$\|\delta^\circ_x[T]\|^2_M \le 4\|C\xi - s\|^2_\Omega, \tag{31}$$

where Ω is the block diagonal matrix whose order equals the dimension of s and whose diagonal blocks corresponding to the coordinates of each $s_{wi}[t]$ and $s_{vi}[t]$, $t \in \mathbb{I}_{N-1}$ in s are Ω_{wi} and Ω_{vi}, respectively.

Proof. Since the consolidated solution satisfies the interconnecting relations specified by H,

$$\begin{aligned} \delta^\circ_w &= \hat{w}^\circ - \hat{w} = \left(\hat{w}^\circ - \bar{w}\right) - \left(\hat{w} - \bar{w}\right) \\ &= H\left(\hat{v}^\circ - \bar{v}\right) - \delta_w = H\left(\hat{v}^\circ - \hat{v}\right) + H\left(\hat{v} - \bar{v}\right) - \delta_w \\ &= H\delta^\circ_v + H\delta_v - \delta_w. \end{aligned} \tag{32}$$

If the subsystems are incrementally dissipative, by adding the inequality (25) with t taking values from 0 to any $T-1 \in \mathbb{I}_{N-1}$ for all subsystems, the deviations should satisfy

$$\sum_{t=0}^{T-1}\sum_{i=1}^{n} \|\delta^\circ_{vi}[t]\|^2_{\Omega_{vi}} \le \sum_{t=0}^{T-1}\sum_{i=1}^{n} \|\delta^\circ_{wi}[t]\|^2_{\Omega_{wi}} - \sum_{i=1}^{n} \|\delta^\circ_{xi}[N]\|^2_{M_i}, \tag{33}$$

where the meaning of δ°_{xi} is analogous to that of δ°_{wi} or δ°_{vi}. Dropping the last term, we have

$$\sum_{t=0}^{T-1} \|\delta^\circ_v[t]\|^2_{\Omega_v} \le \sum_{t=0}^{T-1} \|\delta^\circ_w[t]\|^2_{\Omega_w}, \tag{34}$$

Substituting (32) into above, we have

$$\begin{aligned} &\textstyle\sum_{t=0}^{T-1} \|\delta^\circ_v[t]\|^2_{\Omega_v} \le \sum_{t=0}^{T-1} \|H\delta^\circ_v[t] + (H\delta_v[t] - \delta_w[t])\|^2_{\Omega_w} \\ &\le 2\left(\textstyle\sum_{t=0}^{T-1} \|H\delta^\circ_v[t]\|^2_{\Omega_w} + \sum_{t=0}^{T-1} \|H\delta_v[t] - \delta_w[t]\|^2_{\Omega_w}\right). \end{aligned} \tag{35}$$

In the above inequality, $2\|H\delta^\circ_v[t]\|^2_{\Omega_w} = \delta^\circ_v[t]^\top (2H^\top \Omega_w H)\delta^\circ_v[t]$ and $\|\delta^\circ_v[t]\|^2_{\Omega_v} = \delta^\circ_v[t]^\top \Omega_v \delta^\circ_v[t]$. Moving the first term on the right-hand side to the left gives

$$\sum_{t=0}^{T-1} \|\delta^\circ_v[t]\|^2_{\Omega_v - 2H^\top \Omega_w H} \le 2\sum_{t=0}^{T-1} \|H\delta_v[t] - \delta_w[t]\|^2_{\Omega_w}, \tag{36}$$

where $\|\cdot\|_{\Omega_v - 2H^\top \Omega_w H}$ is well-defined due to (30). This implies that δ°_v remains bounded. Then, from (32), we obtain the bound of δ°_w:

$$\begin{aligned} &\textstyle\sum_{t=0}^{T-1} \|\delta^\circ_w[t]\|^2_{\Omega_w} = \sum_{t=0}^{T-1} \|H\delta^\circ_v[t] + H\delta_v[t] - \delta_w[t]\|^2_{\Omega_w} \\ &\le 2\textstyle\sum_{t=0}^{T-1} \left(\|\delta^\circ_v[t]\|^2_{H^\top \Omega_w H} + \|H\delta_v[t] - \delta_w[t]\|^2_{\Omega_w}\right) \end{aligned} \tag{37}$$

which implies that

$$\begin{aligned} &\textstyle\sum_{t=0}^{T-1} \left(\|\delta^\circ_w[t]\|^2_{\Omega_w} - \|\delta^\circ_v[t]\|^2_{\Omega_v}\right) \\ &\le \textstyle\sum_{t=0}^{T-1} \left(2\|\delta^\circ_v[t]\|^2_{H^\top \Omega_w H} + 2\|H\delta_v[t] - \delta_w[t]\|^2_{\Omega_w} - \|\delta^\circ_v[t]\|^2_{\Omega_v}\right) \\ &= \textstyle\sum_{t=0}^{T-1} \left(-\|\delta^\circ_v[t]\|^2_{\Omega_v - 2H^\top \Omega_w H} + 2\|H\delta_v[t] - \delta_w[t]\|^2_{\Omega_w}\right) \\ &\le 2\textstyle\sum_{t=0}^{T-1} \|H\delta_v[t] - \delta_w[t]\|^2_{\Omega_w}. \end{aligned}$$

(38)

Hence with the incremental dissipativity properties of the subsystems, the deviation of the future states in the consolidated solution from the predicted solution is bounded as

$$\begin{aligned}\|\delta_x^\circ[T]\|_M^2 &\le \textstyle\sum_{t=0}^{T-1}\left(\|\delta_w^\circ[t]\|_{\Omega_w}^2 - \|\delta_v^\circ[t]\|_{\Omega_v}^2\right)\\ &\le 2\textstyle\sum_{t=0}^{T-1}\|H\delta_v[t]-\delta_w[t]\|_{\Omega_w}^2\\ &\le 4\textstyle\sum_{t=0}^{T-1}\left(\|\delta_v[t]\|_{H^\top\Omega_w H}^2 + \|\delta_w[t]\|_{\Omega_w}^2\right).\end{aligned} \tag{39}$$

Since $\Omega_v \succeq 2H^\top\Omega_w H \succeq H^\top\Omega_w H$,

$$\|\delta_x^\circ[T]\|_M^2 \le 4\sum_{t=0}^{T-1}\left(\|\delta_v[t]\|_{\Omega_v}^2 + \|\delta_w[t]\|_{\Omega_w}^2\right). \tag{40}$$

By further relaxing the right-hand side to a summation from 0 to $N-1$, we obtain (31). □

Next we consider the effect of $\delta_x^\circ = \hat{x}^\circ - \hat{x}$ on the stability of the consolidated solution. Denote the deviation of the total objective function value by $\delta_V^\circ := \hat{V}^\circ - \hat{V}$, where $\hat{V}$ is the objective function value of the prediction and $\hat{V}^\circ$ is that of the consolidated solution.

Lemma 4 (Effect of violations to the interconnecting relations on the objective function value of the consolidated solution)**.** *Suppose that the subsystems are incrementally dissipative (according to Assumption 4) with (30) and that the MPC formulation satisfies Assumptions 2 and 5. When distributed optimization of the MPC problem gives a solution (ξ, s) and corresponding subsystem prediction with objective function value $\hat{V}$, the consolidated solution will have an objective function of $\hat{V}^\circ = \hat{V} + \delta_V^\circ$, where*

$$|\delta_V^\circ| \le r_1\hat{V}^{1/2}\|C\xi - s\| + r_2\|C\xi - s\|^2 \tag{41}$$

for some positive constants r_1 and r_2.

Proof. Assumption 5 implies that the values of the terms in the objective function have the following bounded deviations:

$$|\delta_{\ell i}^\circ| := |\ell_i(\hat{x}_i^\circ, \hat{u}_i) - \ell_i(\hat{x}_i, \hat{u}_i)| \le l_i\|\hat{x}_i\|\cdot\|\delta_{xi}^\circ\| + \frac{l_i}{2}\|\delta_{xi}^\circ\|^2 \tag{42}$$

where the time index $t \in \mathbb{I}_{N-1}$ is omitted. The deviation in the terminal cost then satisfies

$$|\delta_{\ell fi}^\circ| \le l_{fi}\|\hat{x}_i[N]\|\cdot\|\delta_{x_i}^\circ[N]\| + \tfrac{l_{fi}}{2}\|\delta_{x_i}^\circ[N]\|^2. \tag{43}$$

Adding up all the stage cost and terminal cost deviations in all the subsystems, we obtain that δ_V° should be bounded by

$$\begin{aligned}|\delta_V^\circ| \le\ & \textstyle\sum_{i=1}^n\Big[\sum_{t=0}^{N-1}\left(\ell_i\|\hat{x}_i^\circ[t]\|\cdot\|\delta_{x_i}^\circ[t]\| + \tfrac{l_i}{2}\|\delta_{x_i}^\circ[t]\|^2\right)\\ &+ l_{fi}\|\hat{x}_i[N]\|\cdot\|\delta_{x_i}^\circ[N]\| + \tfrac{l_{fi}}{2}\|\delta_{x_i}^\circ[N]\|^2.\end{aligned} \tag{44}$$

Using the Cauchy-Schwarz inequality, we further relax the above inequality as

$$\begin{aligned}|\delta_V^\circ| \le\ & \left[\textstyle\sum_{i=1}^n\left(\sum_{t=0}^{N-1} l_i^2\|\hat{x}_i^\circ[t]\|^2 + l_{fi}^2\|\hat{x}_i^\circ[N]\|^2\right)\right]^{1/2}\|\delta_x^\circ\|\\ &+ \max_{i\in\mathbb{I}_n^+}\max\left(\tfrac{l_i}{2}, \tfrac{l_{fi}}{2}\right)\|\delta_x^\circ\|^2.\end{aligned} \tag{45}$$

From (40), we have

$$\lambda_{\min}(M)\|\delta_x^\circ[T]\|^2 \le 4\lambda_{\max}(\Omega)\|C\xi - s\|^2 \tag{46}$$

for any $T \in \mathbb{I}_{N-1}$. Hence

$$\|\delta_x^\circ\|^2 \le 4N\frac{\lambda_{\max}(\Omega)}{\lambda_{\min}(M)}\|C\xi - s\|^2. \tag{47}$$

Under Assumption 5, choose sufficiently large positive constants r_1 and r_2, e.g., according to

$$\begin{aligned}r_1 &\ge 2\max_{i\in\mathbb{I}_n^+}\max\left(l_i^2/g_i, l_{fi}^2/g_{fi}\right)\cdot N^{1/2}\lambda_{\max}^{1/2}(\Omega)/\lambda_{\min}^{1/2}(M),\\ r_2 &\ge 4\max_{i\in\mathbb{I}_n^+}\max\left(l_i, l_{fi}\right)\cdot N\lambda_{\max}(\Omega)/\lambda_{\min}(M).\end{aligned} \tag{48}$$

Then the conclusion is proved. □

4.3. The Lyapunov envelope algorithm

Motivated by the conclusion of Lemma 4, we consider the following modified formulation of the MPC problem:

$$\begin{aligned}\min_{\xi,s}\ & \textstyle\sum_{i=1}^n\left(\phi_i(\xi_i) + p_1\|C_i\xi_i - s_i\| + p_2\|C_i\xi_i - s_i\|^2\right)\\ \text{s.t.}\ & \xi_i \in \Xi_i,\ i \in \mathbb{I}_n^+\\ & s = [s_1; \ldots; s_n] \in \Sigma.\end{aligned} \tag{49}$$

Suppose that in an iterative algorithm to solve (49), starting from an initial solution of (ξ^0, s^0) satisfying $C_i\xi_i^0 = s_i^0$, $i \in \mathbb{I}_n^+$, the iterations from $(\xi^r, s^r) \to (\xi^{r+1}, s^{r+1})$ are such that the objective function is monotonically non-increasing. If such an algorithm can be performed in a distributed manner on an agent-coordinator architecture, i.e., by iterations of an agent update step and a coordinating step, then regardless of whether the algorithm finds or converges to a point (ξ, s) satisfying $s = C\xi$, the intermediate solution terminated after any iteration will guarantee that the consolidated solution will have a decreased objective function without the penalty terms. For convenience, we introduce the following definition.

Definition 1 (Lyapunov envelope and Lyapunov envelope algorithm)**.** Under the conditions of Lemma 4, the objective function of (49) is called a Lyapunov envelope. An algorithm for solving (49) that initializes at a feasible point (ξ, s) satisfying $s = C\xi$ and returns a feasible solution with the objective function not exceeding that of the initial point is called a Lyapunov envelope algorithm.

Specifically, the modified FBS algorithm discussed in Section 3.2, where each iteration comprises of (21) and (22) with the penalty chosen as $\varpi(\hat{s} - \bar{s}) = p_1\|\hat{s} - \bar{s}\| + p_2\|\hat{s} - \bar{s}\|^2$, is a Lyapunov descent algorithm of solving (49) simply in a block coordinate descent paradigm. In the agent update step (21), each subsystem i in parallel updates a $\xi_i \in \Xi_i$ that reduces or minimizes the corresponding term in the objective function of (49), i.e.,

$$\xi_i^{r+1} \in \arg\min_{\xi_i\in\Xi_i}\left(\phi_i(\xi_i) + p_1\|C_i\xi_i - s_i^r\| + p_2\|C_i\xi_i - s_i^r\|^2\right). \tag{50}$$

In the coordinating step, s is updated by minimizing the objective terms associated with s in (49), i.e.,

$$s^{r+1} \in \arg\min_{s\in\Sigma}\left(p_1\|C\xi^{r+1} - s\| + p_2\|C\xi^{r+1} - s\|^2\right). \tag{51}$$

Formally, the algorithm is summarized in Algorithm 1 .

```
for t = 0, 1, ... do
    Initialize: ξ^0 and s^0 satisfying Cξ^0 = s^0;
    Set: r ← 0;
    while r ≤ r_max do
        for i = 1, ..., n in parallel do
            Solve ξ_i^{r+1} from (50);
        end
        Solve s^{r+1} from (51);
        r ← r + 1;
    end
end
```

Algorithm 1: Lyapunov envelope algorithm using block coordinate descent.

Now consider the distributed MPC at time $k+1$. If the consolidated solution of the previous execution (under $x[t]$) is assigned as the initial solution, clearly, it satisfies the interconnecting relations and hence is itself a consolidated solution. As the iteration proceeds, the intermediate solutions yield a monolithically non-increasing sequence of upper bounds on the control cost when

they become consolidated (Lyapunov envelope). By the same reasoning as in Section 2, the Lyapunov envelope can be used as a control-Lyapunov function, which, under the same appropriate assumptions, leads to the asymptotic closed-loop stability property.

However, the relaxation of interconnecting relations $s = C\xi$ as soft constraints may imply possible violation of state constraints $\mathbb{X}$ and $\mathbb{X}_f$ in the consolidated solutions and hence recursive feasibility. To resolve this issue, we denote by $V_{LE}(x)$ the optimized objective function value under the Lyapunov envelope when the states are sampled at x and $V^{\kappa_f}(x)$ the objective function value when initialized under the baseline controller κ_f ($V_{LE}(x) \le V^{\kappa_f}$). Let $\mathbb{F}$ be the region on which the centralized MPC problem is feasible (so that at any $x \in \mathbb{F}$, the subproblems can always be feasibly solved), and $a^* > 0$ be the maximum one among such values of $a > 0$ such that the sublevel set $\mathbb{S}(V^{\kappa_f}, a) = \{x | V^{\kappa_f}(x) \le a\}$ is a subset of $\mathbb{F}$. Then, assuming that the trajectory starts on $\mathbb{S}(V^{\kappa_f}, a^*)$, the trajectory is bound in $\mathbb{S}(V^{\kappa_f}, a^*) \subseteq \mathbb{F}$, and hence the recursive feasibility is preserved.

Therefore, we have reached our main conclusion.

Theorem 2 (Closed-loop stability guarantee of the Lyapunov envelope algorithm)**.** *Suppose that the following propositions hold:*

1. *Assumptions 1, 2, 4 and 5 hold for the system dynamics and MPC formulation;*
2. *In every execution of distributed MPC, the initial solution is set as the consolidated solution of the previous execution, extended under κ_f (according to (10));*
3. *A Lyapunov envelope algorithm is used for distributed MPC, where the penalty coefficients p_1 and p_2 are such that $r_1 = p_1/(a^*)^{1/2}$ and $r_2 = p_2$ satisfy (48).*

Then starting on $\mathbb{S}(V^{\kappa_f}, a^*)$, the closed-loop system is asymptotically stable under the Lyapunov envelope algorithm.

Comparing the Lyapunov envelope algorithm to the FBS discussed in Section 3.1, the following advantages can be noted.

1. The coordinator does not need the information of the feasible regions of interconnecting variables s_i for every subsystem, and does not restrict itself to solutions that must make all subsystems match themselves to.
2. The coordinator does not need to evaluate the gradient of φ, which is implicitly expressed in terms of the optimal values of the subsystems' optimization problems.
3. The iterations to coordinate the subsystems can be early terminated. Despite the violations of interconnecting relations, the consolidated intermediate solutions are of controlled quality in the sense of a well-bounded control cost.
4. The Lyapunov envelope is monotonic with iterations, eliminating the possibility that intermediate consolidated solutions may result in deteriorated performance. The user may therefore conveniently find a trade-off between computational efficiency and solution optimality.

4.4. Remarks on the Lyapunov envelope algorithm

In the remainder of this session, we provide some remarks to facilitate a better understanding and practical utilization of the Lyapunov envelope algorithm.

Remark 1 (Substituting. ℓ_2-norm with ℓ_1-norm) In the coordinating step of each iteration, the s variable is updated under a given ξ by solving the following problem:

$$\begin{aligned} \min_s \quad & p_1 \|s - C\xi\| + p_2 \|s - C\xi\|^2 \\ \text{s.t.} \quad & s = [s_1; \dots; s_n] \in \Sigma, \end{aligned} \tag{52}$$

which is a convex optimization problem. By using an auxiliary scalar σ with constraints $\|s - C\xi\|_\Omega \le \sigma$, the above formulation is transformed into a second-order conic programming (SOCP) problem.

In fact, due to the equivalence of ℓ_q-norms for all $q \ge 1$, one can substitute the ℓ_2 penalty $\|s - C\xi\|_2$ with any other ℓ_q forms. Especially, it is desirable to substitute it with the ℓ_1-norm for the benefit of reformulating the problem into a quadratic programming form (QP):

$$\begin{aligned} \min_{s,\varsigma} \quad & p_1 e^\top \varsigma + p_2 \|s - C\xi\|^2 \\ \text{s.t.} \quad & -\varsigma \le s - C\xi \le \varsigma, \ H_s s = 0, \end{aligned} \tag{53}$$

where e is a vector with all components equal to 1, and $H_s s = 0$ stands for the interconnecting constraints (arising from $w = H\nu$) in $s \in \Sigma$. The ℓ_1-norm substitution is desirable also for the subsystems to update ξ since with the auxiliary vector ς introduced, the non-smoothness of the ℓ_1-norm term can be resolved.

Remark 2 (Sharp Lagrangean and exact penalty in optimization)**.** The formality of the penalty terms in the Lyapunov envelope algorithm has an interesting conceptual connection with the studies of augmented Lagrangean duality in optimization theory, which investigate the conditions under which the dual variables support exact penalty representations. Specifically, to minimize an objective function f under constraints $c(x) = 0$ and $x \in \mathbb{X}$, if minimizing the augmented Lagrangean

$$L(x, \lambda, \rho) = f(x) + \lambda^\top c(x) + \rho \varpi(c(x)) \tag{54}$$

(where ϖ is called a penalty function) under given λ and a finite value of ρ gives exactly the optimum as the original problem, then λ is said to support an exact penalty representation and hence the augmented Lagrangean can result in zero duality gap (Rubinov et al., 1999; Huang and Yang, 2003). In the afore-mentioned literature, a sharp Lagrangean, namely an augmented Lagrangean with $\varpi(\cdot) = \|\cdot\|_q$ instead of a quadratic penalty $\varpi(\cdot) = \|\cdot\|^2$ was proposed.

It was proved that in order to have $\lambda = 0$ supporting an exact penalty representation, the ℓ_q-norm is chosen so that the order $q > 0$ correctly reflects the sensitivity of the objective function to the violations to the constraints $c(x) = 0$ (Huang and Yang, 2003, Theorem 4.6). In this sense, the penalty terms in the Lyapunov envelope play the role of such exact penalty functions to account for the effect of disturbances (i.e., violations to the interconnecting relations among subsystems) on the objective function. The choice of a combination of a sharp and classical form with both ℓ_1 and quadratic penalty takes into account the first-order and second-order disturbance effects, which is implied by the continuity conditions (Assumption 5).

Yet, different from the context of augmented Lagrangean duality, the Lyapunov envelope algorithm does not consider a full optimization of the augmented Lagrangean, which is computationally expensive. Instead, we adopt a distributed approach to successively improve the objective value. While it does not guarantee a convergence to the optimum, the control performance can be made satisfactory with appropriate parameters p_1 and p_2.

Remark 3 (Parameter tuning and scalings)**.** The choice of parameters p_1 and p_2 has a fundamental effect on the computational efficiency and the control performance of the resulting solution. On one hand, according to the previous discussions, p_1 and p_2 should overestimate the first-order and second-order effects of disturbances on the objective function for maintaining the closed-loop stability. On the other hand, if p_1 and p_2 are too large (over-conservative), then the ill-conditioned proportion between the original objective and the penalty terms will slow down the iterations to improve the solutions and largely increase the time for numerical computations. Hence the two parameters should be tuned just large enough to yield closed-loop stability.

Within the two parameters, p_1 should be considered as more important than p_2, since the squared ℓ_2-norm is dominated by the ℓ_1-norm when $C\xi - s$ is close to 0 and thus it is the ℓ_1 penalty that mainly determines the control performance. Therefore to tune the algorithm, the value of p_1 should be decided primarily for a trade-off between control and computational performances, followed by an appropriate choice of p_2. In this paper we follow this empirical tuning procedure. As a future direction, it may be possible to choose the parameters based on a systematic and rigorous dissipativity analysis of system dynamics.

It is worth noting that although formally we can reduce the number of parameters down to 2, the relative scalings of the components of $C\xi - s$ are intangibly crucial, since well-scaled penalty terms reduce the conversativeness of estimating the disturbance effects. Without a-priori scalings, one can determine the scaling factors of $C\xi - s$, denoted by θ, using the information of static gain G from disturbances to the states (or outputs) and scaling factors η of the states (or outputs) in the objective function:

$$\min_{\eta} \|G\theta - \eta\|^2. \tag{55}$$

Remark 4 (Satisfaction of incremental dissipativity)**.** For linear systems, supposing that the interconnecting inputs and outputs w and v are well-scaled, one may verify the LMI (28) with simple forms of Ω_{wi} and Ω_{vi} for each subsystem, e.g., $\Omega_{wi} = \beta_{wi} I$, $\Omega_{vi} = \beta_{vi} I$ for β_{wi}, $\beta_{vi} > 0$ to confirm that the subsystems are incrementally dissipative. For uncertain and nonlinear systems, it is usually difficult to directly verify the incremental dissipativity assumption. Nevertheless, it can be argued that incremental dissipativity is a property that holds for many real-world systems, especially chemical processes whose dynamics should obey the first and second laws of thermodynamics.

To obtain the dissipativity properties from the dynamic models of chemical processes, conditions and procedures of thermodynamic analysis have been discussed, e.g., in Alonso and Ydstie (1996) and Hangos et al. (2001). Recently, data-driven methods of estimating dissipativity properties using trajectory samples from simulations or plant tests (e.g., Koch et al., 2020) have also been proposed, which may help to reduce the difficulty of a rigorous model-based thermodynamic analysis.

Remark 5 (Enforcement of incremental dissipativity)**.** The condition on the incremental dissipativity of the interconnecting subsystems $\Omega_v - 2H^\top \Omega_w H \succ 0$ intrinsically requires that the subsystems must be sufficiently self-stabilizing, so that the effect of hypothetical disturbances (δ_w, δ_v) on the interconnected system is not "snowballed".

If the condition $\Omega_v - 2H^\top \Omega_w H \succ 0$ is not satisfied because of the insufficient dissipativity of some subsystems, it is desirable to design for these subsystems an auxiliary controller κ_i to shape their dissipativity properties prior to implementing distributed MPC and impose distributed MPC based on such an auxiliary controller. For example, if ramp dynamics exist in some subsystems, proportional feedback can be first established from each ramp variable to one of its manipulated inputs. This essentially amounts to a variable transformation $u_i = \pi_i(x_i) + u_i'$, where π_i is the auxiliary controller, u_i represents the original inputs, and u_i' is the transformed inputs for MPC.

For a systematic design of this auxiliary controller π_i, if needed, one can seek to optimize an L_2 or dissipativity characterization of the effect of w_i as exogenous disturbances on v_i regarded as controlled outputs. For linear systems or systems whose open-loop dissipative behavior is known, the controller synthesis problem to shape the closed-loop dissipativity has been well addressed (Haddad et al., 1994; Willems and Trentelman, 2002). In the context of systems for which dissipativity can be learned in a data-driven manner, the dissipativity learning control framework integrates the learning of open-loop dissipativity from data and the controller synthesis step (Tang and Daoutidis, 2019a; 2021a).

Remark 6 (Effect of decompositions)**.** The performance of distributed MPC depends on the choice of decomposition, i.e., the number of subsystems and the allocation of variables into these subsystems that determine the subsystem model and the interconnecting relations. The natural idea that subsystems should have significant weaker couplings between them than the relations inside them has motivated network-theoretic methods of generating such decompositions (Daoutidis et al., 2018; 2019). The Lyapunov envelope algorithm proposed in the present paper imposes the following requirements on the decomposition: (i) all the subsystems should be incrementally dissipative so that the effect of interconnecting inputs w_i on the interconnecting outputs v_i should be small, (ii) the couplings should be weak enough so that the interconnected subsystems have finite response over disturbances on the interconnections. Thus, in addition to considering the number of connections inside and across the subsystems as in community detection, a truly high-quality decomposition should be such that the parameters p_1 and p_2 as defined in (48) that capture the plantwide effect of disturbances can be made small.

5. Case study

5.1. System description

We use the vinyl acetate monomer chemical plant of Chen et al. (2003) to carry out a case study of the application of the Lyapunov envelope algorithm to plantwide distributed MPC of large-scale systems. A detailed description of the nonlinear first-principles continuous-time dynamics and source codes can be found at McAvoy's repository (http://terpconnect.umd.edu/mcavoy/VAC20Material/). In this work, we only use a linearized, forward-discretized dynamics derived from the first-principles model and neglect the plant-model mismatch as well as possible disturbances and noises. In practice, the plantwide MPC of process systems is mostly based on an identified linear dynamics (with nonlinearities appended in an ad-hoc way), and the resulting deviations from the model are usually accounted for by calibrating the measurements against a disturbance model, which is known as offset-free MPC (see, e.g., Pannocchia and Rawlings, 2003) and is beyond the scope of this paper.

The entire plant has 246 state variables (x), 21 inputs (u) and 43 outputs (y), and divided into 3 subsystems. Subsystem 1 (acetic acid storage tank, vaporizer and reactor) has 83 states, 7 inputs, and 14 outputs; subsystem 2 (feed effluent heat exchange, phase separator, absorption column and CO_2 removal) has 93 states, 9 inputs and 17 outputs; subsystem 3 (distillation column) has 70 states, 6 inputs, and 12 outputs. The pattern of interactions among subsystems can be seen in Fig. 3, where the structure of the linearized model matrix with the rows and columns permutated in the order of subsystems 1, 2 and 3, namely

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \begin{bmatrix} A_{11} & A_{12} & A_{13} & B_{11} & B_{12} & B_{13} \\ A_{21} & A_{22} & A_{23} & B_{21} & B_{22} & B_{23} \\ A_{31} & A_{32} & A_{33} & B_{31} & B_{32} & B_{33} \\ C_{11} & C_{12} & C_{13} & D_{11} & D_{12} & D_{13} \\ C_{21} & C_{22} & C_{23} & D_{21} & D_{22} & D_{23} \\ C_{31} & C_{32} & C_{33} & D_{31} & D_{32} & D_{33} \end{bmatrix}, \tag{56}$$

is plotted in a color map. To account for the interactions among the 3 subsystems, 60 interconnection variables (w, v) are created, with the dimensions of w being 12, 23, 25 and the dimensions of v being 23, 33, 4, respectively.

The linear discrete-time dynamics of the system represented by the matrix A is examined. Except for 11 eigenvalues that are close

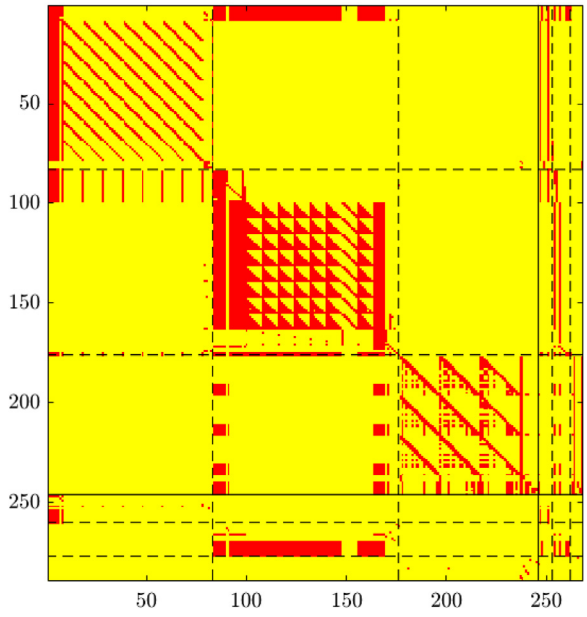

Fig. 3. Model structure of the vinyl acetate monomer system. Yellow (light) and red (dark) pixels stand for zero and nonzero entries, respectively. Solid lines separate blocks of A, B, C, D, and dashed lines separate subsystems. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

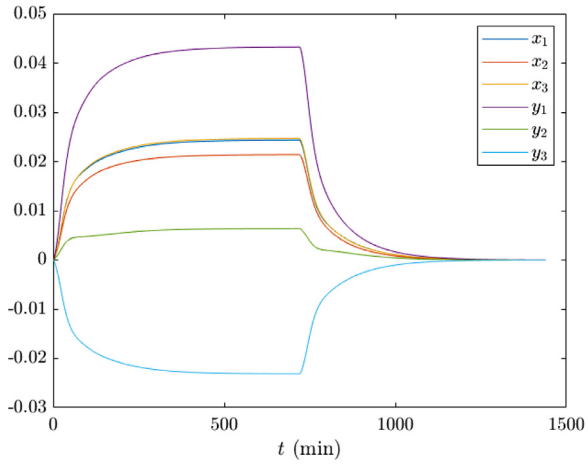

Fig. 4. Closed-loop behavior under centralized LQR represented by the response of separator compositions to distillation reflux ratio changes.

to 1 (due to ramp dynamics), the remaining 235 eigenvalues reside in the interior of the unit disk of the complex plane, implying that the open-loop system is stable if the ramp dynamics are eliminated. As suggested in Chen et al. (2003), feedback from the liquid levels in the process to a corresponding number of selected inputs is adopted to move these poles inside the unit disk. We also specify the scalings of the inputs and outputs as in Chen et al. (2003).

Under $Q = 100C^\top C$, $R = I$, a centralized LQR controller is designed and the closed-loop stability is verified through simulation. For example, Fig. 4 shows the response of the 6 molar fractions of the 3 species in the two phases of the separator to a 25% step increase in the distillation column reflux ratio at 0 min and its restoration to the original value at 720 min. Hence, we use this centralized LQR controller as the base controller κ_f in MPC. The value of the cost function under κ_f is thus a function of the initial point at the sampling point. The monotonic decrease of such a function $V^{\kappa_f}(x)$ along the closed-loop trajectories will be exploited to demonstrate the convergence behaviors under different MPC schemes.

Table 1
Process constraints considered in the case study.

Variables	Bounds
Oxygen composition in the gas recycle	$\leq$ 8 mol%
Pressure in the gas recycle	$\leq$ 140 psia
Peak reactor temperature	$\leq$ 200 °C
Liquid levels in all the units	$\geq$ 10%, $\leq$ 90%
Reactor feed temperature	$\geq$ 130 °C
Hot effluent temperature of the feed-effluent heat exchanger	$\geq$ 130 °C
Acetic acid composition in the decanter organic phase	$\leq$ 0.06 mol%
Vinyl acetate composition in the column bottom	$\leq$ 0.01 mol%

Table 2
Sizes of the MPC subproblems.

Subsystem	# Variables	# Constraints
1	31,403	36,803
2	41,673	49,233
3	26,350	28,150

5.2. Distributed MPC setup

Due to the difficulty of optimization on a plantwide first-principles nonlinear model, we use linearized model prediction for distributed MPC, with quadratic forms as control costs for the subsystems: $\ell_i(x_i, u_i) = x_i^\top Q_i x_i + u_i^\top R_i u_i$. Weightings of the inputs and states are the same as described in the previous subsection for decentralized LQR, and the constraint sets $\mathbb{X}$, $\mathbb{X}_f = \mathbb{X}$, $\mathbb{U}$ are defined as the ranges specified in Tables 2 and 4 of Chen et al. (2003). For the absent values, we supplement with a range of ± 1 times their respective scaling factors. Additional process constraints are imposed according to the control objectives 1–8 in Chen et al. (2003), summarized in Table 1. The terminal cost is set as $\ell_{fi} = x_i^\top S_i x_i$, where S_i is solved from decentralized LQR. For simplicity, we assume that all state variables are directly measurable, while in reality, a Kalman filter is usually needed for state observation. The sampling time is 1 min, and a prediction horizon of 180 min (empirically tuned to achieve closed-loop stability under centralized MPC) is used. For the illustration of the proposed method, we only consider a regulating control scenario without complexities such as exogenous disturbances or transitions between different operating regions.

We first test the setup of distributed MPC with a decentralized MPC counterpart (without any coordination and assuming the interconnecting inputs of all the subsystems are $w = 0$). The initial condition is a different steady state with the reflux ratio of the distillation column is 25% higher than the steady state to be regulated at. It was found that the decentralized MPC does not guarantee closed-loop stability. While it is possible that in practice, in the presence of disturbance calibration for offset-free MPC, the closed-loop stability can be recovered, the loss of stability under a non-calibrated decentralized MPC strategy indicates that the interactions among the subsystems are non-trivial, for which it is desirable to consider coordination.

We then compare decentralized MPC to a fully centralized MPC with the same objective function and constraints. The centralized MPC problem at each sampling time is a QP with 56,946 variables and 61,626 constraints. Fig. 5 and Fig. 6 show the trajectories of the input and output variables under centralized MPC during a simulation of 360 min. On average, it takes 9.525 seconds to solve each centralized MPC problem, which occupies 15.87% of the sampling interval. Suppose that, for example, the user needs the computational time to be no greater than 10% of the sampling time; then centralized MPC for this application is not well implementable in

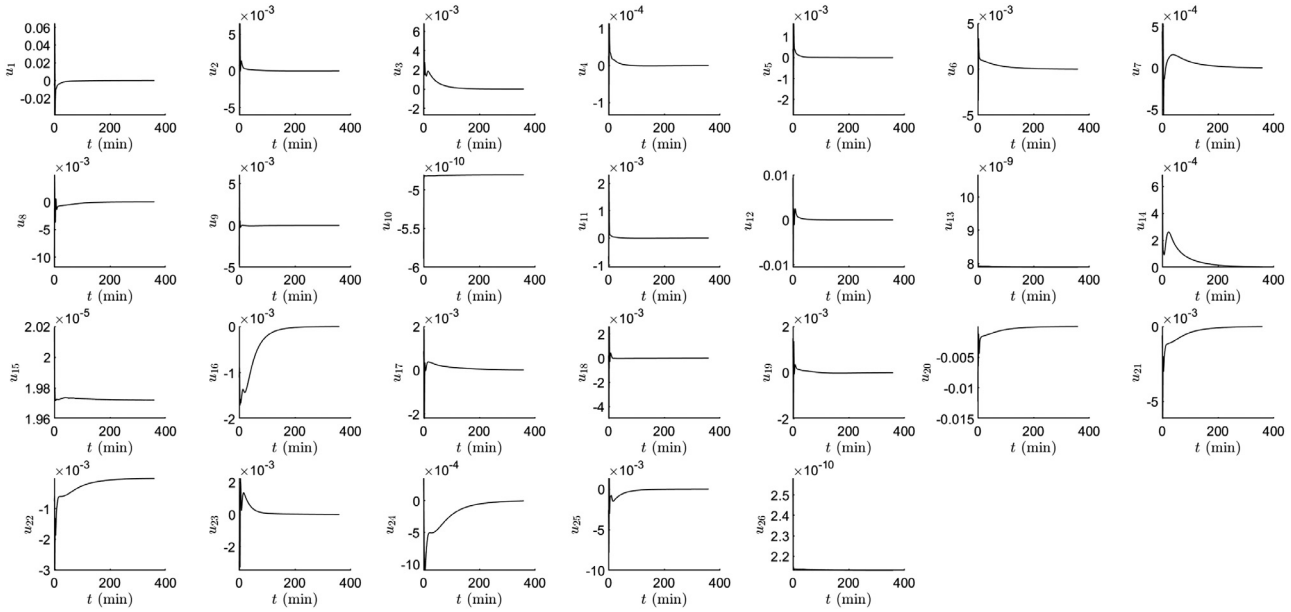

Fig. 5. Trajectories of inputs under centralized MPC.

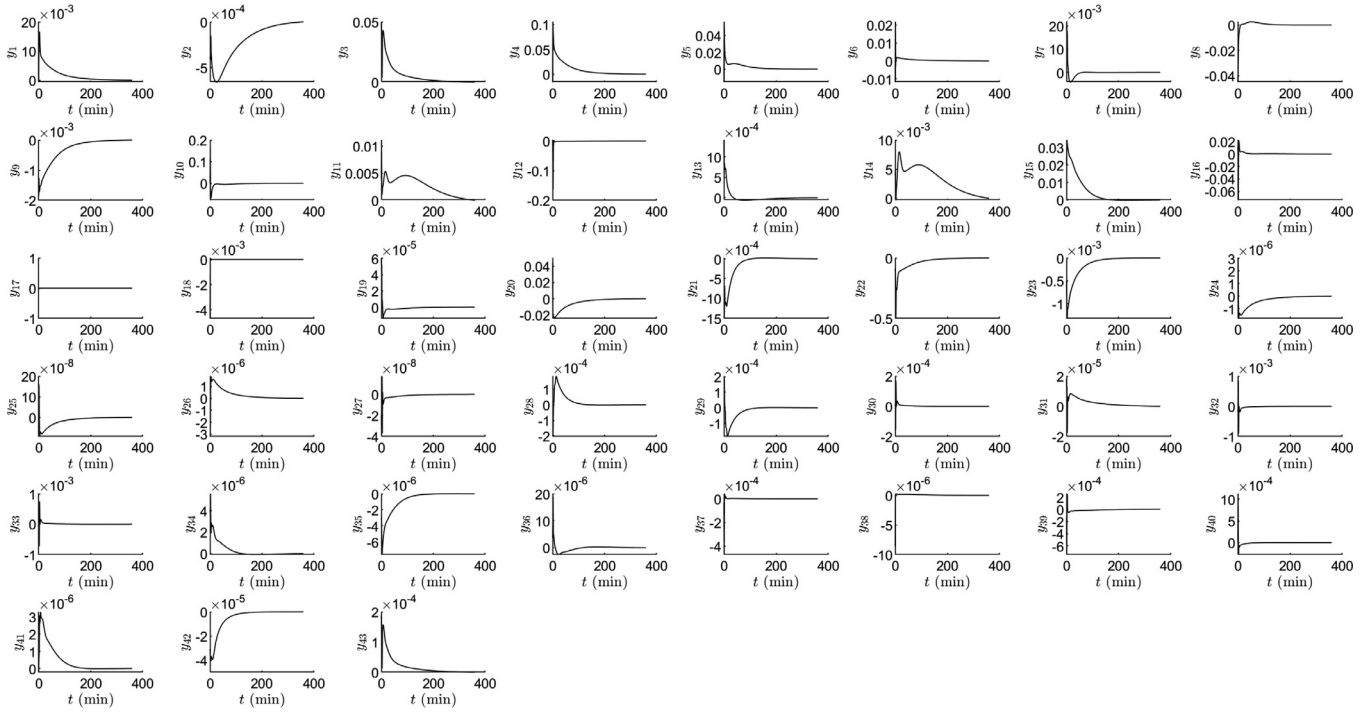

Fig. 6. Trajectories of the outputs under centralized MPC.

real time. This motivates the adoption of Lyapunov envelope algorithm.

5.3. Results

Using distributed MPC, the number of variables and constraints for the MPC subproblems of the 3 subsystems are given in Table 2. Each subsystem has a subproblem much smaller than the centralized MPC problem. However, due to the newly added interconnecting variables w and v among the subsystems, the total sizes of the 3 subproblems are larger than that of the centralized MPC formulation. We note that the subsystem models here are constructed rigorously by splitting a linearized and discretized first-principles model. In a more practical setting where the model is obtained through system identification procedures, it can be reasonably expected that the weak first-principle relations among the subsystems will not appear in the model, and hence the inflation of subproblem sizes will be much more reduced.

For the Lyapunov envelope algorithm, we set $p_1 = 10$, $p_2 = 10$ and at each sampling time, the number of iterations performed before termination is increased from 1 to 3.

Fig. 7 illustrates the resulting control and computational performance in terms of the trajectory of $V^{\kappa_f}(x)$ and the accumulated computational time with increasing simulation time. It can be observed that the trajectories of $V^{\kappa_f}(x)$ under 1, 2 and 3 iterations per sampling time are very close, which is also close to the trajectory

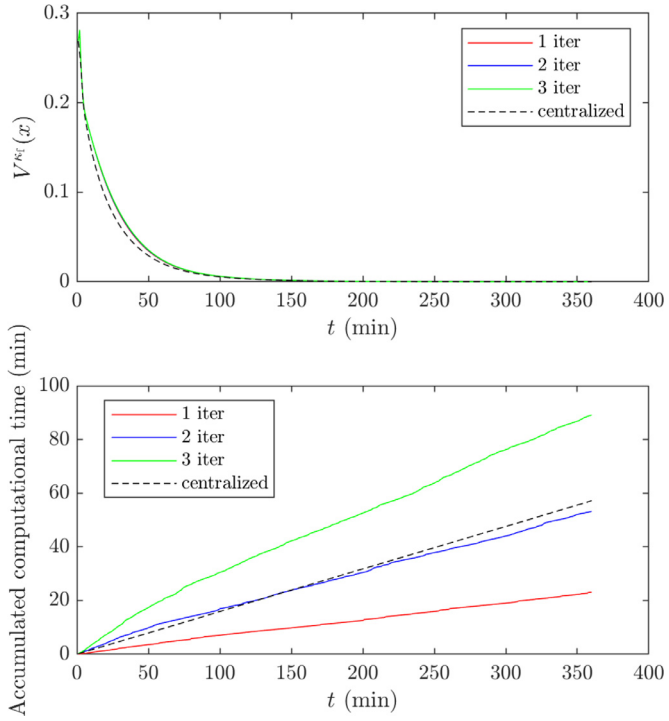

Fig. 7. Control and computational performance of distributed MPC with the Lyapunov envelope algorithm ($p_1 = 10$, $p_2 = 10$).

Table 3
Control performance under different parameter settings.

$p_1 \backslash p_2$	1	3	10	30	100
1	7.7553	7.7297	7.7313	7.7646	7.8164
3	7.6139	7.5890	7.6162	7.6295	7.6255
10	6.2868	6.2841	6.3212	6.2693	6.3228
30	5.6693	5.6718	5.6682	5.6725	5.6678
100	5.6641	5.6634	5.6633	5.6660	5.6635

under centralized MPC. This implies that under the current parameter settings, using 1 iteration per sampling time is sufficient. On the other hand, if more iterations are used, the average computational time per sampling will increase. For our decomposition into 3 subsystems only, in order to maintain computational time lower than that of centralized MPC, at most 2 iterations should be used at each sampling time. By using the 1-iteration Lyapunov envelope algorithm, the average computational time is reduced to 3.850 seconds per sampling time (6.42% of the sampling interval). Compared to centralized MPC, the computational time is accelerated by a factor of 2.47.

The proposed algorithm is dependent on the ℓ_1 and quadratic penalty parameters p_1 and p_2. We vary each of these two parameters among 5 values (1, 3, 10, 30, 100), with 25 combinations, and record the resulting control performance in terms of the accumulated value of $\sum_t V^{\kappa_f}(x(t))$ along the simulated trajectory (lower values indicate better control performance). The results are shown in Table 3.

As expected (see Remark 3), we observe that p_1 has a more significant effect on the control performance than p_2 does. As p_1 increases, the control performance is improved and approaches that of the centralized MPC, for which the corresponding index is 5.6500. Under a fixed $p_2 = 10$ and different values of p_1, we compare the trajectories of $V^{\kappa_f}(x)$ throughout the simulation in Fig. 8. It can be seen that the centralized MPC has the steepest descent of the cost function, and when p_1 is above 30, the trajectory is almost overlapping with that of centralized MPC. When p_1 takes a smaller value, the relaxation of interconnecting relations gives rise to disturbances that slow down the descent.

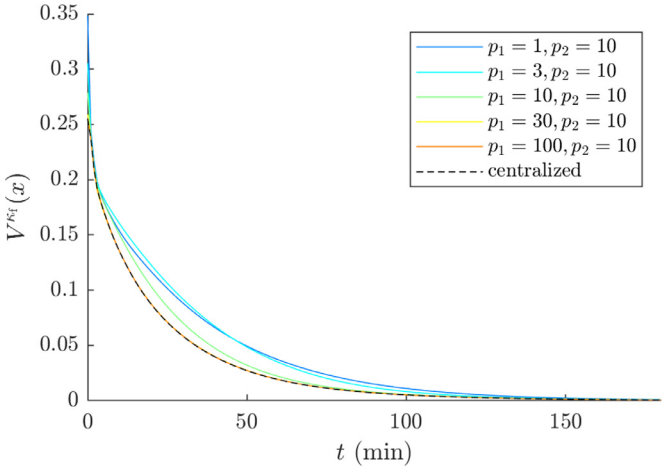

Fig. 8. Comparison of control performance under different parameter tuning in the Lyapunov envelope algorithm.

Table 4
Computational performance under different parameter settings.

$p_1 \backslash p_2$	1	3	10	30	100
1	6.9143	6.4046	7.3793	6.8260	8.5807
3	6.4241	5.1308	5.7656	5.3364	7.1228
10	4.0851	4.0367	3.8473	3.9245	5.3450
30	5.5038	10.1044	26.2289	4.3044	19.4970
100	9.7516	6.9934	16.3448	34.1468	23.5383

The corresponding computational performance in terms of the average computational time per sampling is shown in Table 4.

The results exhibit a tendency that over-conservative large values of p_1 and p_2 lead to high computational expenses and should be avoided. Low parameter values do not always favor better computational efficiency, either, due to the over-relaxation of interconnecting relations and hence too large update steps. The optimal computational performance is achieved at around $p_1 = 10$ and $p_2 = 10$, at which the control performance is also satisfactorily close to that of the centralized MPC.

6. Conclusions

In this paper we focus on the problem of real-time implementability of coordination in distributed MPC strategies, and propose a Lyapunov envelope algorithm that allows the coordination scheme to be early terminated after an arbitrary number of iterations without loss of closed-loop stability and yields solution of improved performance if more iterations are allowed. The algorithm is essentially a primal block coordinate descent one, where the interconnecting relations among the subsystems are relaxed into a combination of ℓ_1 and quadratic penalty terms.

Central to the penalty formulation is the concept of Lyapunov envelope. For the intermediate solution of the subsystems during iterations, for which the interconnecting relations are relaxed and hence violated, the Lyapunov envelope accounts for the effect of such violations on the control objective of the resulting consolidated solution, hence establishing itself as a control-Lyapunov function, assuming that the subsystems are sufficiently incrementally dissipative. Therefore, we conclude in simple words that *when the subsystems are sufficiently self-stabilizing and well decoupled, early termination can be allowed in the presence of ℓ_1 and quadratic penalties, and therefore coordination can be easily implemented in real-time on a plantwide scale.*

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Wentao Tang: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft. **Prodromos Daoutidis:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing.

Acknowledgment

This work was supported by NSF-CBET. NSF Grant CBET-1926303

References

Alonso, A.A., Ydstie, B.E., 1996. Process systems, passivity and the second law of thermodynamics. Comput. Chem. Eng. 20, S1119–S1124.
Baldea, M., Daoutidis, P., 2007. Control of integrated process networks – a multi–time scale perspective. Comput. Chem. Eng. 31 (5-6), 426–444.
Biegler, L.T., Zavala, V.M., 2009. Large-scale nonlinear programming using IPOPT: an integrating framework for enterprise-wide dynamic optimization. Comput. Chem. Eng. 33 (3), 575–582.
Bolte, J., Sabach, S., Teboulle, M., 2014. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Prog. 146 (1-2), 459–494.
Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trend. Mach. Learn. 3 (1), 1–122.
Brogliato, B., Lozano, R., Maschke, B., Egeland, O., 2020. Dissipative Systems Analysis and Control: Theory and Applications, third ed. Springer.
Cai, X., Tippett, M.J., Xie, L., Bao, J., 2014. Fast distributed MPC based on active set method. Comput. Chem. Eng. 71, 158–170.
Chen, G.H., Rockafellar, R.T., 1997. Convergence rates in forward–backward splitting. SIAM J. Optim. 7 (2), 421–444.
Chen, R., Dave, K., McAvoy, T.J., Luyben, M., 2003. A nonlinear dynamic model of a vinyl acetate process. Ind. Eng. Chem. Res. 42 (20), 4478–4487.
Christofides, P.D., Scattolini, R., Muñoz de la Peña, D., Liu, J., 2013. Distributed model predictive control: a tutorial review and future research directions. Comput. Chem. Eng. 51, 21–41.
Daoutidis, P., Tang, W., Allman, A., 2019. Decomposition of control and optimization problems by network structure: concepts, methods and inspirations from biology. AIChE J. 65 (10), e16708.
Daoutidis, P., Tang, W., Jogwar, S.S., 2018. Decomposing complex plants for distributed control: perspectives from network theory. Comput. Chem. Eng. 114, 43–51.
Duchi, J., Singer, Y., 2009. Efficient online and batch learning using forward backward splitting. J. Mach. Learn. Res. 10, 2899–2934.
Eckstein, J., Bertsekas, D.P., 1992. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Prog. 55 (1–3), 293–318.
Farokhi, F., Shames, I., Johansson, K.H., 2014. Distributed MPC via dual decomposition and alternative direction method of multipliers. In: Distributed Model Predictive Control Made Easy. Springer, pp. 115–131.
Ferranti, L., Keviczky, T., 2015. A parallel dual fast gradient method for MPC applications. In: Proc. 54th Conf. Decis. Control (CDC). IEEE, pp. 2406–2413.
Fiacco, A.V., Ishizuka, Y., 1990. Sensitivity and stability analysis for nonlinear programming. Ann. Oper. Res. 27 (1), 215–235.
Ghadimi, E., Teixeira, A., Shames, I., Johansson, M., 2014. Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems. IEEE Trans. Autom. Control 60 (3), 644–658.
Giselsson, P., Boyd, S., 2016. Linear convergence and metric selection for Douglas-Rachford splitting and ADMM. IEEE Trans. Autom. Control 62 (2), 532–544.
Giselsson, P., Rantzer, A., 2013. On feasibility, stability and performance in distributed model predictive control. IEEE Trans. Autom. Control 59 (4), 1031–1036.
Haddad, W.M., Bernstein, D.S., Wang, Y.W., 1994. Dissipative H_2/H_∞ controller synthesis. IEEE Trans. Autom. Control 39 (4), 827–831.
Hangos, K.M., Bokor, J., Szederkényi, G., 2001. Hamiltonian view on process systems. AIChE J. 47 (8), 1819–1831.
Houska, B., Frasch, J., Diehl, M., 2016. An augmented Lagrangian based algorithm for distributed nonconvex optimization. SIAM J. Optim. 26 (2), 1101–1127.
Huang, X.X., Yang, X.Q., 2003. A unified augmented Lagrangian approach to duality and exact penalization. Math. Oper. Res. 28 (3), 533–552.
Krishnamoorthy, D., Foss, B., Skogestad, S., 2019. A primal decomposition algorithm for distributed multistage scenario model predictive control. J. Process Control 81, 162–171.
Kumar, A., Christofides, P.D., Daoutidis, P., 1998. Singular perturbation modeling of nonlinear processes with nonexplicit time-scale multiplicity. Chem. Eng. Sci. 53 (8), 1491–1504.
Lions, P.-L., Mercier, B., 1979. Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. 16 (6), 964–979.
Lopez-Negrete, R., DAAmato, F.J., Biegler, L.T., Kumar, A., 2013. Fast nonlinear model predictive control: formulation and industrial process applications. Comput. Chem. Eng. 51, 55–64.
Luyben, M.L., Tyreus, B.D., Luyben, W.L., 1997. Plantwide control design procedure. AIChE J. 43 (12), 3161–3174.
Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M., 2000. Constrained model predictive control: stability and optimality. Automatica 36 (6), 789–814.
Morari, M., Arkun, Y., Stephanopoulos, G., 1980. Studies in the synthesis of control structures for chemical processes: Part I: formulation of the problem. process decomposition and the classification of the control tasks. analysis of the optimizing control structures. AIChE J. 26 (2), 220–232.
Morari, M., Lee, J.H., 1999. Model predictive control: past, present and future. Comput. Chem. Eng. 23 (4–5), 667–682.
Mota, J.F., Xavier, J.M., Aguiar, P.M., Püschel, M., 2014. Distributed optimization with local domains: applications in MPC and network flows. IEEE Trans. Autom. Control 60 (7), 2004–2009.
Negenborn, R.R., Maestre, J.M., 2014. Distributed model predictive control: an overview and roadmap of future research opportunities. IEEE Control Syst. Mag. 34 (4), 87–97.
Pannocchia, G., Rawlings, J.B., 2003. Disturbance models for offset-free model-predictive control. AIChE J. 49 (2), 426–437.
Patterson, M.A., Rao, A.V., 2014. GPOPS-II: a MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming. ACM Trans. Math. Softw. (TOMS) 41 (1), 1–37.
Pavlov, A., Marconi, L., 2008. Incremental passivity and output regulation. Syst. Control Lett. 57 (5), 400–409.
Price, R.M., Georgakis, C., 1993. Plantwide regulatory control design procedure using a tiered framework. Ind. Eng. Chem. Res. 32 (11), 2693–2705.
Pu, Y., Zeilinger, M.N., Jones, C.N., 2014. Inexact fast alternating minimization algorithm for distributed model predictive control. In: Proc. 53rd Conf. Decis. Control (CDC). IEEE, pp. 5915–5921.
Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Eng. Pract. 11 (7), 733–764.
Quirynen, R., Di Cairano, S., 2020. PRESAS: block-structured preconditioning of iterative solvers within a primal active-set method for fast model predictive control. Optim. Control Appl. Meth. 41 (6), 2282–2307.
Rawlings, J.B., Mayne, D.Q., Diehl, M.M., 2017. Model Predictive Control: Theory, Computation, and Design, second ed. Nob Hill Publishing.
Rubagotti, M., Patrinos, P., Bemporad, A., 2014. Stabilizing linear model predictive control under inexact numerical optimization. IEEE Trans. Autom. Control 59 (6), 1660–1666.
Rubinov, A.M., Glover, B.M., Yang, X., 1999. Decreasing functions with applications to penalization. SIAM J. Optim. 10 (1), 289–313.
Scattolini, R., 2009. Architectures for distributed and hierarchical model predictive control – a review. J. Process Control 19 (5), 723–731.
Skogestad, S., 2004. Control structure design for complete chemical plants. Comput. Chem. Eng. 28 (1–2), 219–234.
Stephanopoulos, G., 1983. Synthesis of control systems for chemical plants a challenge for creativity. Comput. Chem. Eng. 7 (4), 331–365.
Tang, W., Daoutidis, P., 2019. Dissipativity learning control (DLC): a framework of input–output data-driven control. Comput. Chem. Eng. 130, 106576.
Tang, W., Daoutidis, P., 2019. Distributed nonlinear model predictive control through accelerated parallel ADMM. In: Am. Control Conf.. IEEE, pp. 1406–1411.
Tang, W., Daoutidis, P., 2021. Dissipativity learning control (DLC): theoretical foundations of input–output data-driven model-free control. Syst. Control Lett. 147, 104831.
Tang, W., Daoutidis, P., 2021. Fast and stable nonconvex constrained distributed optimization: the ELLADA algorithm. Optim. Eng. 22. In press, doi:10.1007/s11081-020-09585-w
Themelis, A., Stella, L., Patrinos, P., 2018. Forward-backward envelope for the sum of two nonconvex functions: further properties and nonmonotone line-search algorithms. SIAM J. Optim. 28 (3), 2274–2303.
Wang, Z., Ong, C.-J., 2018. Accelerated distributed MPC of linear discrete-time systems with coupled constraints. IEEE Trans. Autom. Control 63 (11), 3838–3849.
Willems, J.C., Trentelman, H.L., 2002. Synthesis of dissipative systems using quadratic differential forms: Part I. IEEE Trans. Autom. Control 47 (1), 53–69.
Wolf, I.J., Marquardt, W., 2016. Fast NMPC schemes for regulatory and economic NMPC – a review. J. Process Control 44, 162–183.
Yang, T., Yi, X., Wu, J., Yuan, Y., Wu, D., Meng, Z., Hong, Y., Wang, H., Lin, Z., Johansson, K.H., 2019. A survey of distributed optimization. Annu. Rev. Control 47, 278–305.
Yu, T., Zhao, J., Xu, Z., Chen, X., Biegler, L.T., 2019. Sensitivity-based hierarchical distributed model predictive control of nonlinear processes. J. Process Control 84, 146–167.
Koch, A., Berberich, J., Köhler, J., Allgöwer, F., 2020. Determining optimal input-output properties: a data-driven approach. arXiv preprint arXiv:2002.03882.