



Instant distributed MPC with reference governor

Martin Figura, Lanlan Su, Masaki Inoue & Vijay Gupta

To cite this article: Martin Figura, Lanlan Su, Masaki Inoue & Vijay Gupta (2024) Instant distributed MPC with reference governor, International Journal of Control, 97:4, 662-672, DOI: [10.1080/00207179.2023.2164868](https://doi.org/10.1080/00207179.2023.2164868)

To link to this article: <https://doi.org/10.1080/00207179.2023.2164868>



Published online: 13 Jan 2023.



Submit your article to this journal [↗](#)



Article views: 290



View related articles [↗](#)



View Crossmark data [↗](#)



Instant distributed MPC with reference governor

Martin Figura^a, Lanlan Su^b, Masaki Inoue^c and Vijay Gupta^a

^aDepartment of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA; ^bDepartment of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, United Kingdom; ^cDepartment of Applied Physics and Physico-Informatics, Keio University, Kanagawa, Japan

ABSTRACT

Model predictive control is a popular choice for systems that must satisfy prescribed constraints on states and control inputs. Although much progress has been made in distributed model predictive control, existing algorithms tend to be computationally expensive. This limits their use in systems with fast dynamics. In this paper, we propose a new distributed model predictive control algorithm that we term as instant distributed model predictive control (iDMPC). The proposed algorithm employs a realisation of the primal-dual algorithm in the controller dynamics for fast computation. We show that the closed-loop system trajectories with the proposed iDMPC algorithm converge asymptotically to a desired reference. To ensure the satisfaction of the state constraints during the transient, we also include an explicit reference governor in the feedback loop.

ARTICLE HISTORY

Received 7 April 2022

Accepted 28 December 2022

KEYWORDS

Distributed model predictive control; suboptimal control; reference governor; dissipativity

1. Introduction

Model predictive control (MPC) is a powerful and elegant control design method especially for nonlinear processes with constraints on states and inputs. Although the method has been applied to a variety of practical applications, it is well-acknowledged that its application to processes with fast dynamics continues to lag. This is primarily due to the fact that solving an optimisation problem at every time step as proposed in MPC may require considerable computational time, which may render the implementation of MPC in systems with fast dynamics unstable (Wang & Boyd, 2009). Consequently, a popular research direction has been to design ways to accelerate the solution of the optimisation problems in MPC. In this direction, we can point to explicit MPC for linear systems (Rubagotti et al., 2014; Tøndel et al., 2003), which relies on an offline computation of quadratic programme (QP) and application of the control input via lookup tables; warm-starting (Richter et al., 2009; Yildirim & Wright, 2002), which takes advantage of the control input computed for a previous time step to initialise the computation for the current time step; and early termination of the optimisation process with a suboptimal control input (Wang & Boyd, 2009).

A recent approach along these lines has been suboptimal MPC, which applies time distributed optimisation to compute control trajectories (Liao-McPherson et al., 2020; Skibik & Nicotra, 2022). Suboptimal MPC adopts the optimisation dynamics from MPC, but it implements the control input while the optimal control problem is being solved. A general suboptimal MPC controller is depicted in Figure 1. The updates performed by suboptimal MPC are fast enough that the dynamic controller can be approximated as a continuous-time dynamical

system. This allows suboptimal MPC to be implemented in systems with fast dynamics provided that the data sampling rate is fast as well. The optimisation dynamics in suboptimal MPC can be selected with a considerable degree of freedom (Hauswirth et al., 2020). In particular, primal-dual gradient flow algorithms have been employed in Yoshida et al. (2019), Nicotra et al. (2018) and Adegbege (2022). An input-to-state stability approach to stabilise trajectories around a reference point is introduced in Nicotra et al. (2018). The control methods proposed in Yoshida et al. (2019) and Adegbege (2022) are based on dissipativity analyses. The former approach involves offline tuning of parameters to ensure asymptotic stability of a reference while the latter is augmented by anti-windup feedback control in the MPC computation. In this work, we look to apply some of these ideas to establish a suboptimal MPC method for distributed systems.

For control of distributed systems, various formulations of distributed model predictive control (DMPC) have been proposed (Camponogara et al., 2002; Keviczky et al., 2006; Negenborn & Maestre, 2014; Richards & How, 2007). Such a DMPC formulation is a natural fit for a variety of problems such as building energy management (Patel et al., 2016), chemical processes (Pourkargar et al., 2017), power generation (Venkat et al., 2008), water supply management (Leirens et al., 2010), or formation control (Liu et al., 2018). Note that the issue of the computational cost of the optimisation algorithm imposing a limit on how fast the dynamics of the process to be controlled can be is even more salient in DMPC. For example, a DMPC scheme was proposed to stabilise a multi-vehicle formation in Dunbar and Murray (2006). In that algorithm, each agent receives the optimal predicted input and state trajectories from its neighbours at every sampling instant, based on which it

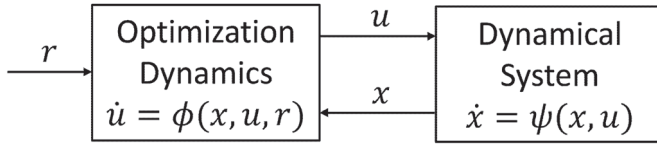


Figure 1. Suboptimal MPC in a feedback loop with a dynamical system.

computes its own predicted trajectories over a receding horizon. To ensure stability, the algorithm requires that the difference between the optimal predicted state trajectories computed at two consecutive sampling times must remain sufficiently small. One can mitigate the gap by increasing the sampling frequency which in turn may turn too small for the DMPC controller to solve the entire optimisation problem. In this paper, we merge the ideas of suboptimal MPC and DMPC to alleviate this problem.

A drawback of the suboptimal MPC schemes (even in the centralised case) is that they only guarantee feasible solutions in a steady state (Jokic & Lazar, 2009). The evolution of state and input trajectories in transient may lead to constraint violation because the applied suboptimal input is not a minimiser of the constrained optimisation problem. In fact, the applied input may not even be feasible (Hauswirth et al., 2020). One way to ensure that the input and state trajectories remain feasible in transient as well is to implement a reference governor (RG) in the feedback loop. For a detailed overview of the RGs and their applications, we refer the reader to the surveys (Garone et al., 2017; Kolmanovsky et al., 2014) and references therein. The RG is a dynamical system that was proposed for tracking problems as an alternative to MPC both in the centralised (Bemporad, 1998; Gilbert et al., 1995) and distributed setting (Casavola et al., 2014; Tedesco et al., 2012). Whereas MPC generates a control input to steer system states to a terminal set by solving an expensive constrained optimisation problem (Limón et al., 2008), the state feedback-RG scheme steers the modified reference trajectory to the desired reference and stabilises the system around the modified reference while satisfying constraints, all being done under significantly lower computational requirements. The computational footprint of the RGs can be reduced further by shifting away from optimisation-based RG schemes to the explicit RG (ERG) (Garone & Nicotra, 2015). The ERG takes advantage of invariant sets, which are strategically aligned with the constraint boundaries, to appropriately adjust the speed of convergence to the desired reference which ensures that the constraints are not violated in the case of convex feasible sets (Garone et al., 2018). We mention that most of the ERG developments have been made for centralised control. The implementation of the ERG in the distributed setting is more challenging as we need to account for communication constraints. Finally, we note that distributed control using the ERG was presented in Nicotra et al. (2015) and that suboptimal MPC with the ERG in the centralised setting was presented in Nicotra et al. (2018).

In this work, we introduce instant distributed MPC (iDMPC) with ERG. To the best of our knowledge, iDMPC is the first suboptimal DMPC method designed to regulate constrained distributed systems. Furthermore, the inclusion of the ERG in

the control synthesis ensures that constraints are satisfied at all times. Our technical results include a proof that under the assumption of no communication delays, iDMPC computes state and input trajectories equivalent to centralised instant MPC as studied in Yoshida et al. (2019), a proof of asymptotic stability using a dissipativity-based analysis, and a guarantee that the constraints are satisfied during the transient phase as well. We demonstrate via simulations that iDMPC has a significantly lower computational burden than the standard distributed MPC and thus can be applied to systems with faster dynamics. Some results in this paper were presented in a preliminary form in the conference paper (Figura et al., 2020). The main novelty over that version is the consideration of state constraints in the dynamics which requires the use of an ERG to ensure feasibility during the transient domain. The proofs have also been expanded and clarified and a more complex simulation example has been considered.

The paper is organised as follows. The problem statement for general distributed MPC is presented in Section 2. The iDMPC algorithm, stability analysis, and ERG are introduced in Section 3. In Section 4, we present simulation results of systems with coupled cost and coupling constraints. The conclusion is provided in Section 5.

Notation

Vectors are expressed in a bold font. The k th element of a vector or the k th row of a matrix is denoted as $[\cdot]_k$. The element of a matrix associated with i th row and j th column is given as $[\cdot]_{ij}$. The interior of a set is denoted as $\text{Int}(\cdot)$, $\mathbf{1}$ is the vector of ones, and $\circ$ is the Hadamard product.

2. Problem statement

2.1 Plant dynamics

We consider linear time-invariant systems with coupled cost and constraints. The continuous-time (CT) dynamics of subsystem i , $i = 1, \dots, M$, are given as

$$\dot{\mathbf{x}}_i = A_{ci}\mathbf{x}_i + B_{ci}\mathbf{u}_i, \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^{m_i}$ and $\mathbf{u}_i \in \mathbb{R}^{n_i}$ are the state and control input, respectively. The dynamics of the interconnected system can be written in a compact form as

$$\dot{\mathbf{x}} = A_c\mathbf{x} + B_c\mathbf{u}, \quad (2)$$

where $\mathbf{u} = [\mathbf{u}_1^\top \dots \mathbf{u}_M^\top]^\top \in \mathbb{R}^n$, $\mathbf{x} = [\mathbf{x}_1^\top \dots \mathbf{x}_M^\top]^\top \in \mathbb{R}^m$, $A_c = \text{diag}(A_{c1}, \dots, A_{cM}) \in \mathbb{R}^{m \times m}$, $B_c = \text{diag}(B_{c1}, \dots, B_{cM}) \in \mathbb{R}^{m \times n}$, $m = \sum_{i=1}^M m_i$, and $n = \sum_{i=1}^M n_i$.

Assumption 2.1: The pair (A_c, B_c) is controllable.

The assumption of controllability enables the system regulation from any state configuration, which is especially important when constraints are imposed on the state trajectories. Since MPC relies on periodically solving an optimisation problem to calculate the control input, it is natural to also consider a discrete-time (DT) formulation of the system dynamics in (1).

We let τ denote the sampling time. The DT dynamics of subsystem i are given as

$$\mathbf{x}_i^\tau = A_i \mathbf{x}_i + B_i \mathbf{u}_i, \quad (3)$$

where $\mathbf{x}_i^\tau = \mathbf{x}_i(t + \tau)$, $A_i = e^{A_{ci}\tau}$, $B_i = \int_0^\tau e^{A_{ci}\xi} d\xi B_{ci}$. Then, the dynamics of the interconnected system are given by

$$\mathbf{x}^\tau = A\mathbf{x} + B\mathbf{u}, \quad (4)$$

where $A = \text{diag}(A_1, \dots, A_M)$ and $B = \text{diag}(B_1, \dots, B_M)$.

2.2 Objective

The objective of our work is to asymptotically stabilise an interconnected system with dynamics defined in (2) at a reference point (the origin) under the least expensive control strategy that ensures that state constraints imposed on individual systems are satisfied. In the following lines, we formulate this objective formally. The formulation is provided in discrete time as it is a conventional approach in solving MPC problems, where non-zero computational time to solve an optimisation problem must be considered (Wang & Boyd, 2009).

We succinctly write the state and input trajectories of subsystem i over an N -step horizon as

$$\mathbf{z}_i = [(\mathbf{u}_i^0)^\top \dots (\mathbf{u}_i^{(N-1)\tau})^\top (\mathbf{x}_i^\tau)^\top \dots (\mathbf{x}_i^{N\tau})^\top]^\top. \quad (5)$$

Furthermore, we write $\mathbf{z} = [\mathbf{z}_1^\top \dots \mathbf{z}_M^\top]^\top \in \mathbb{R}^{N(n+m)}$ to denote the vector with state and input trajectories over an N -step horizon of all M subsystems. Even though the trajectory $\mathbf{z}$ is time-dependent, we do not include time in the notation for simplicity. Moreover, we let the subscript $-i$ denote all subsystems that are coupled with subsystem i through cost or constraints.

A key component of any optimisation-based controller is a cost functional that evaluates the cost incurred by applying a sequence of control inputs over a horizon of chosen length. In our distributed setting, we consider a local stage cost

$$c_i(\mathbf{x}_i, \mathbf{z}_{-i}, \mathbf{u}_i) = c_{x,i}(\mathbf{x}_i, \mathbf{z}_{-i}) + c_{u,i}(\mathbf{u}_i), \quad (6)$$

where the term $c_{x,i}(\cdot, \cdot)$ corresponds to a coupled stage cost and the term $c_{u,i}(\cdot)$ denotes the stage cost incurred by applying a local control input. To evaluate the cost of subsystem i over an N -step horizon, we define the functional

$$f_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{u}_i) = \sum_{j=0}^{N-1} c_i(\mathbf{x}_i^{j\tau}, \mathbf{z}_{-i}^{j\tau}, \mathbf{u}_i^{j\tau}). \quad (7)$$

Finally, the cost incurred by the interconnected system over an N -step horizon is

$$f(\mathbf{z}, \mathbf{x}) = \sum_{i=1}^M f_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{u}_i). \quad (8)$$

For notational simplicity, we do not consider a terminal cost in the cost functional, even though it is a common practice in MPC (Stewart et al., 2010). An extension of the cost functional that includes the terminal cost is straightforward. Either way, the cost functional $f(\mathbf{z}, \mathbf{x})$ has to satisfy the following assumption.

Assumption 2.2: The cost functional $f(\mathbf{z}, \mathbf{x})$ is strongly convex and continuously differentiable, and $\nabla f(\mathbf{0}, \mathbf{0}) = \mathbf{0}$.

In addition to the coupling between local states through cost, each subsystem is subject to inequality constraints. We characterise two types of inequality constraints:

- (1) the states and control inputs of subsystem i must remain safe on the predicted trajectory, i.e. $\mathbf{x}_i \in \mathcal{X}_i$ and $\mathbf{u}_i \in \mathcal{U}_i$ on the interval $[t, t + N\tau]$, where $\mathcal{X}_i$ and $\mathcal{U}_i$ are safe sets of states and control inputs, respectively;
- (2) the states of individual subsystems must satisfy coupled constraints on the trajectory (e.g. collision constraints).

In our problem formulation, we assume that the subsystems are coupled only through state constraints and that the state and input constraints are decoupled. Hence, we interpret the constraints through vector functions $\mathbf{q}_{x,i}(\mathbf{x}_i, \mathbf{z}_{-i})$ and $\mathbf{q}_{u,i}(\mathbf{u}_i)$ that are evaluated as positive whenever the constraints are violated. We succinctly write $\mathbf{q}_i(\mathbf{x}_i, \mathbf{z}_{-i}, \mathbf{u}_i) = [\mathbf{q}_{x,i}(\mathbf{x}_i, \mathbf{z}_{-i})^\top \mathbf{q}_{u,i}(\mathbf{u}_i)^\top]^\top$ to denote the stage constraint function of subsystem i . To capture all constraint functions of subsystem i on the N -step trajectory, including the terminal state constraint $\mathbf{q}_{x,i}(\mathbf{x}_i^{N\tau}, \mathbf{z}_{-i}^{N\tau})$, we define a vector function

$$\mathbf{g}_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{u}_i) = \begin{bmatrix} \mathbf{q}_i(\mathbf{x}_i^0, \mathbf{z}_{-i}^0, \mathbf{u}_i^0) \\ \vdots \\ \mathbf{q}_i(\mathbf{x}_i^{(N-1)\tau}, \mathbf{z}_{-i}^{(N-1)\tau}, \mathbf{u}_i^{(N-1)\tau}) \\ \mathbf{q}_{x,i}(\mathbf{x}_i^{N\tau}, \mathbf{z}_{-i}^{N\tau}) \end{bmatrix}. \quad (9)$$

For compactness, we also write the constraints imposed on the entire network as

$$\mathbf{g}(\mathbf{z}, \mathbf{x}) = \begin{bmatrix} \mathbf{g}_1(\mathbf{z}_1, \mathbf{z}_{-1}, \mathbf{x}_1, \mathbf{u}_1) \\ \vdots \\ \mathbf{g}_M(\mathbf{z}_M, \mathbf{z}_{-M}, \mathbf{x}_M, \mathbf{u}_M) \end{bmatrix}. \quad (10)$$

The constraint function $\mathbf{g}(\mathbf{z}, \mathbf{x})$ must satisfy the following assumption.

Assumption 2.3: The constraint function $\mathbf{g}(\mathbf{z}, \mathbf{x})$ is convex and continuously differentiable in the first argument.

Next, we define an equality constraint function that includes dynamics of the interconnected system. We introduce matrices H_i and V_i which include linear dynamics of the individual subsystems as follows

$$H_i = \begin{bmatrix} B_i & 0 & \dots & 0 & -I & 0 & \dots & 0 \\ 0 & B_i & \ddots & \vdots & A_i & -I & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 & \ddots & \ddots & 0 \\ 0 & \dots & 0 & B_i & 0 & 0 & A_i & -I \end{bmatrix},$$

$$V_i = [A_i^\top \quad 0 \quad \dots \quad 0]^\top.$$

The equality constraint for subsystem i is given as

$$\mathbf{h}_i(\mathbf{z}_i, \mathbf{x}_i) = H_i \mathbf{z}_i + V_i \mathbf{x}_i, \quad (11)$$

which corresponds to the DT dynamics in (3) on the N -step horizon. The equality constraint for the interconnected system,

induced by its dynamics in (4), is given as

$$\mathbf{h}(\mathbf{z}, \mathbf{x}) = H\mathbf{z} + V\mathbf{x}, \quad (12)$$

where $H = \text{diag}(H_1, \dots, H_M)$ and $V = \text{diag}(V_1, \dots, V_M)$. We make the following assumption about the feasibility and reachability of the origin.

Assumption 2.4: *The origin is a strictly feasible reference point, i.e. $\mathbf{g}(\mathbf{0}, \mathbf{0}) < 0$, and $\mathbf{h}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$. For every $\mathbf{x}_i \in \mathcal{X}_i$, $i = 1, \dots, M$, there exists $\mathbf{u}_i \in \mathcal{U}_i$ which asymptotically stabilises subsystem i at the origin.*

We have established that a subsystem with dynamics in (1) is coupled with its neighbours through the cost in (7) and constraints in (9). From the definitions we have made so far, it is easy to see that each subsystem must have access to the current state and predicted trajectories of all neighbouring subsystems, $\mathbf{x}_{-i}$ and $\mathbf{z}_{-i}$, to evaluate the local cost functional $f_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{x}_{-i})$ and the local coupling constraint $\mathbf{g}_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{x}_{-i})$. Therefore, we need to define a communication graph on which the subsystems share these trajectories. We let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the communication graph, where $\mathcal{V} = \{1, \dots, M\}$ denotes a set of subsystems and $\mathcal{E}$ denotes a set of edges that represent communication links between the subsystems. We assume that agent j transmits its state $\mathbf{x}_j$ and predicted trajectories $\mathbf{z}_j$ to agent i , i.e. $(j, i) \in \mathcal{E}$, if and only if the cost functional $f_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{x}_{-i})$ or the constraint $\mathbf{g}_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{x}_{-i})$ depend on the states of subsystem j . We also let $\mathcal{V}_i$ denote a set of subsystems that are coupled with subsystem i . We recall that the neighbours' signals collectively adopt subscript $-i$ in the notation, e.g. $\mathbf{x}_{-i}$.

In the remainder of this section, we introduce two optimisation problems that we aim to solve. First, we consider the interconnected system as a whole and define a centralised optimisation problem. We note that a centralised controller has access to the state $\mathbf{x}$ and predicted trajectories $\mathbf{z}$. We have defined the cost function in (8), inequality constraints in (10), and equality constraints in (12). The centralised optimisation problem is given as follows:

$$\min_{\mathbf{z}} f(\mathbf{z}, \mathbf{x}) \quad \text{s.t. } \mathbf{g}(\mathbf{z}, \mathbf{x}) \leq 0, \mathbf{h}(\mathbf{z}, \mathbf{x}) = 0. \quad (13)$$

Due to the controllability, (strong) convexity stated in Assumptions 2.1–2.4 and linearity of the system dynamics in (4), the optimisation problem in (13) is convex. Thus, it has a unique optimal point $\mathbf{z}^*$ which satisfies the KKT conditions as follows:

$$\begin{aligned} \nabla f(\mathbf{z}^*, \mathbf{x}) + [\nabla \mathbf{g}(\mathbf{z}^*, \mathbf{x})] \boldsymbol{\mu}^* + [\nabla \mathbf{h}(\mathbf{z}^*, \mathbf{x})] \boldsymbol{\lambda}^* &= 0 \\ \boldsymbol{\mu}^* \geq 0, \quad \mathbf{g}(\mathbf{z}^*, \mathbf{x}) \leq 0, \quad \boldsymbol{\mu}^* \circ \mathbf{g}(\mathbf{z}^*, \mathbf{x}) &= 0, \quad \mathbf{h}(\mathbf{z}^*, \mathbf{x}) = 0. \end{aligned} \quad (14)$$

Here, $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ are Lagrange dual variable vectors associated with inequality and equality constraints, respectively. The predicted state and input trajectory $\mathbf{z}^*$ is a unique optimal trajectory for a given state $\mathbf{x}$. In case $\mathbf{x} = \mathbf{0}$, the corresponding optimal trajectory is $\mathbf{z}^* = \mathbf{0}$. We note that our ultimate goal is to break down the centralised optimisation problem in (13) and have each agent optimise its own predicted state and input trajectories

$\mathbf{z}_i$. Therefore, we define the following distributed optimisation problem:

$$\min_{\mathbf{z}_i} f_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{x}_{-i}) \quad \text{s.t. } \mathbf{g}_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{x}_{-i}) \leq 0, \mathbf{h}_i(\mathbf{z}_i, \mathbf{x}_i) = 0. \quad (15)$$

In the next subsection, we propose iDMPC to solve the distributed optimisation problem in (15). In fact, we take advantage of the structure of the cost functional $f(\mathbf{z}, \mathbf{x})$ and the constraints $\mathbf{g}(\mathbf{z}, \mathbf{x})$ to show that an iDMPC controller solves both the distributed and the centralised optimisation problem. The proposed iDMPC control method is a fast alternative to DMPC that solves (15) as well. We note that traditional DMPC controllers such as Camponogara et al. (2002) solve the optimisation problem within a time step τ . Then, the controller applies the first input of a computed trajectory, $\mathbf{u}_i$, to the dynamical system. In contrast, iDMPC instantly applies the control input $\mathbf{u}_i$ while solving the optimisation problem. It is an optimisation-based feedback control strategy designed to drive system state and input trajectories to the reference (origin) such that the KKT conditions in (14) are satisfied in a steady state.

3. Instant distributed model predictive control

Instant MPC is a suboptimal MPC method that features the primal-dual algorithm in the controller dynamics. The control input applied by an iMPC controller corresponds to the intermediate solution of the gradient-based optimisation of (13). Since the applied control inputs are suboptimal, we are particularly keen to design iDMPC to ensure asymptotic stability of the closed-loop system as well as the convergence of the predicted trajectories. We begin with a definition of the iDMPC algorithm and then proceed with its analysis. In the first part of the analysis, we focus on the computation of the predicted state and input trajectories. We prove that the trajectories generated by iDMPC and iMPC are equivalent under a few assumptions on communication between subsystems. Furthermore, we establish a sufficient condition for the asymptotic stability of the reference point (origin) when a dynamical system is controlled by iDMPC. The iDMPC and iMPC controllers are depicted in Figure 2.

3.1 Instant distributed MPC

In this subsection, we introduce the iDMPC algorithm that solves the distributed optimal control problem in (15). Furthermore, we show that the optimisation performed by iDMPC amounts to solving the centralised optimal control problem (13) under a few assumptions on computation and communication between the subsystems.

To maintain consistency with the definition of the inequality and equality constraints in (10) and (12), we let the associated Lagrange dual variables, $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$, be given as

$$\boldsymbol{\mu} = [\boldsymbol{\mu}_1^\top \quad \dots \quad \boldsymbol{\mu}_M^\top]^\top, \quad \boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^\top \quad \dots \quad \boldsymbol{\lambda}_M^\top]^\top. \quad (16)$$

In addition to the local Lagrange dual variables, we define three design parameters for iDMPC, namely $\alpha > 0$, $\beta \geq 0$, and $\kappa = 1 + 2\alpha\beta$. We recall that optimisation-based controllers apply

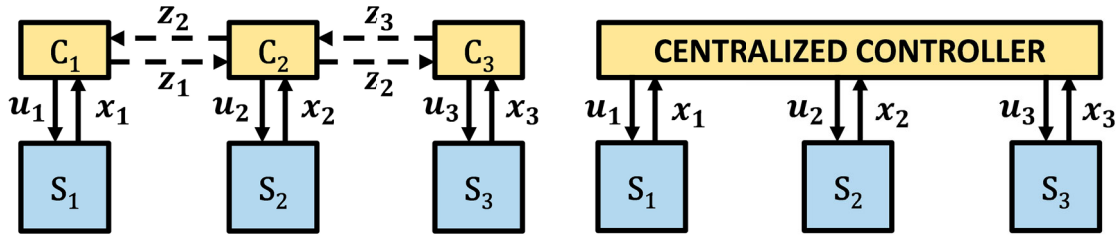


Figure 2. Interconnected systems controlled by iDMPC (left) and iMPC. Physical and communication paths are depicted by solid and dashed lines, respectively.

the first control input of the computed control input sequence to the plant, and hence we let $\mathbf{u}_i = E_i \mathbf{z}_i := [I \ 0 \ \dots \ 0] \mathbf{z}_i$. The inequality constraint function is evaluated (element-wise) as follows

$$[\mathbf{g}_i]_{\mu_i}^+ = \begin{cases} \mathbf{g}_i & \text{if } \mu_i > 0 \\ \max\{\mathbf{g}_i, \mathbf{0}\} & \text{if } \mu_i = 0. \end{cases} \quad (17)$$

Finally, the iDMPC algorithm is given Algorithm 1. For simplicity in the notation, we use shorthand $f_{-i} = \sum_{j \in \mathcal{V}^i} f_j$.

Algorithm 1 iDMPC algorithm

Initial data $\mathbf{z}_i(0), \mu_i(0) \leftarrow \mathbf{0}, \lambda_i(0) \leftarrow \mathbf{0}$
Until convergence
 Measure or receive $\mathbf{x}_i(t^-), \mathbf{x}_{-i}(t^-), \mathbf{z}_{-i}(t^-)$
 Compute
 $\dot{\mathbf{z}}_i(t) = -\nabla_{\mathbf{z}_i}(f_i + f_{-i}) - [\nabla_{\mathbf{z}_i} \mathbf{g}_i] \mu_i - \kappa [\nabla_{\mathbf{z}_i} \mathbf{h}_i](\lambda_i + \beta \dot{\lambda}_i)$
 $\dot{\mu}_i(t) = [\mathbf{g}_i]_{\mu_i}^+$
 $\dot{\lambda}_i(t) = -\alpha(1 + \alpha\beta)^{-1} \lambda_i + (1 + \alpha\beta)^{-1} \mathbf{h}_i$
 Measure $\mathbf{x}_i(t)$
 Send $\mathbf{x}_i(t)$ and $\mathbf{z}_i(t)$ to the neighbours
 Apply $\mathbf{u}_i(t)$

The agents that employ Algorithm 1 utilise only local information from their neighbours, $\mathbf{x}_{-i}$ and $\mathbf{z}_{-i}$, in order to compute their own state and input trajectories $\mathbf{z}_i$. This is a major advantage over centralised iMPC (Yoshida et al., 2019) that collects data from all subsystems and computes input and state trajectories for the entire interconnected system.

Remark 3.1: Algorithm 1 does not handle constraints in a transient state. In Section 3.3, we address this problem by introducing the ERG.

We make the following assumption on the communication between subsystems and some algorithm parameters to make the analysis of the iDMPC algorithm more amenable.

Assumption 3.1: *There is no time delay in the communication, and the time constant τ and the time horizon N are the same for all subsystems.*

In the remainder of this section, we show that iDMPC is equivalent to centralised iMPC (Yoshida et al., 2019) if we assume instant communication on the graph $\mathcal{G}$. However, there is a nuance between the centralised and distributed control

problem. We note that the coupled constraints occur in the vector $\mathbf{g}(\mathbf{z}, \mathbf{x})$ multiple times because they are enforced at different subsystems (e.g. two subsystems try to avoid a collision). By eliminating the duplicate entries in $\mathbf{g}(\mathbf{z}, \mathbf{x})$, we define a new constraint $\mathbf{g}'(\mathbf{z}, \mathbf{x})$. Moreover, we define a Lagrange dual variable vector μ' that is associated with $\mathbf{g}'(\mathbf{z}, \mathbf{x})$. Similarly to Yoshida et al. (2019), the iMPC dynamics are given as follows

$$\dot{\mathbf{z}} = -\nabla f(\mathbf{z}, \mathbf{x}) - [\nabla \mathbf{g}'(\mathbf{z}, \mathbf{x})] \mu' - \kappa [\nabla \mathbf{h}(\mathbf{z}, \mathbf{x})](\lambda + \beta \dot{\lambda}) \quad (18a)$$

$$\dot{\mu}' = [\mathbf{g}'(\mathbf{z}, \mathbf{x})]_{\mu'}^+ \quad (18b)$$

$$\dot{\lambda} = -\alpha(1 + \alpha\beta)^{-1} \lambda + (1 + \alpha\beta)^{-1} \mathbf{h}(\mathbf{z}, \mathbf{x}), \quad (18c)$$

where $[\mathbf{g}'(\mathbf{z}, \mathbf{x})]_{\mu'}^+$ is defined element-wise as follows

$$[\mathbf{g}'(\mathbf{z}, \mathbf{x})]_{\mu'}^+ = \begin{cases} \mathbf{g}'(\mathbf{z}, \mathbf{x}) & \text{if } \mu' > 0 \\ \max\{\mathbf{g}'(\mathbf{z}, \mathbf{x}), \mathbf{0}\} & \text{if } \mu' = 0. \end{cases} \quad (19)$$

Theorem 3.1: *Under Assumption 3.1, the state and input trajectories $\mathbf{z}$ computed by iDMPC (Algorithm 1) are the same as the state and input trajectories $\mathbf{z}$ computed by iMPC in (18).*

Proof: We analyse the dynamics of iMPC and show that they are embedded in iDMPC. First, we consider all terms in (18a). By differentiating $f(\mathbf{z}, \mathbf{x})$ with respect to $\mathbf{z}$, we obtain

$$\nabla f(\mathbf{z}, \mathbf{x}) = \begin{bmatrix} \nabla_{\mathbf{z}_1} f(\mathbf{z}, \mathbf{x}) \\ \vdots \\ \nabla_{\mathbf{z}_M} f(\mathbf{z}, \mathbf{x}) \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{z}_1} (f_1 + f_{-1}) \\ \vdots \\ \nabla_{\mathbf{z}_M} (f_M + f_{-M}) \end{bmatrix}.$$

The second term is distributed as follows

$$[\nabla \mathbf{g}'(\mathbf{z}, \mathbf{x})] \mu' = \begin{bmatrix} \nabla_{\mathbf{z}_1} \mathbf{g}'(\mathbf{z}, \mathbf{x}) \\ \vdots \\ \nabla_{\mathbf{z}_M} \mathbf{g}'(\mathbf{z}, \mathbf{x}) \end{bmatrix} \mu' = \begin{bmatrix} [\nabla_{\mathbf{z}_1} \mathbf{g}_1] \mu_1 \\ \vdots \\ [\nabla_{\mathbf{z}_M} \mathbf{g}_M] \mu_M \end{bmatrix}.$$

Next, we recall from (12) that H and V are block diagonal matrices. Therefore,

$$\nabla \mathbf{h}(\mathbf{z}, \mathbf{x}) = \text{diag}(\nabla_{\mathbf{z}_1} \mathbf{h}_1(\mathbf{z}_1, \mathbf{x}_1), \dots, \nabla_{\mathbf{z}_M} \mathbf{h}_M(\mathbf{z}_M, \mathbf{x}_M)),$$

where $\mathbf{h}_i(\mathbf{z}_i, \mathbf{x}_i)$ is defined in (11). We define $\lambda' = \lambda + \beta \dot{\lambda}$. Analyzing the terms in (18a), we obtain

$$\begin{aligned} \dot{\mathbf{z}} &= -\nabla f(\mathbf{z}, \mathbf{x}) - [\nabla \mathbf{g}'(\mathbf{z}, \mathbf{x})] \mu' - \kappa [\nabla \mathbf{h}(\mathbf{z}, \mathbf{x})] \lambda' \\ &= - \begin{bmatrix} \nabla_{\mathbf{z}_1} (f_1 + f_{-1}) + [\nabla_{\mathbf{z}_1} \mathbf{g}_1] \mu_1 + \kappa [\nabla_{\mathbf{z}_1} \mathbf{h}_1] \lambda'_1 \\ \vdots \\ \nabla_{\mathbf{z}_M} (f_M + f_{-M}) + [\nabla_{\mathbf{z}_M} \mathbf{g}_M] \mu_M + \kappa [\nabla_{\mathbf{z}_M} \mathbf{h}_M] \lambda'_M \end{bmatrix} \end{aligned}$$

$$= [\dot{\mathbf{z}}_1^\top \quad \dots \quad \dot{\mathbf{z}}_M^\top]^\top.$$

Second, we analyse (18b). Since $\mathbf{g}(\mathbf{z}, \mathbf{x})$ contains duplicate entries of $\mathbf{g}'(\mathbf{z}, \mathbf{x})$, the dynamics of $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ are the same, except for the dimension. Third, we analyse (18c) to obtain

$$\begin{aligned} \dot{\boldsymbol{\lambda}} &= -\alpha(1 + \alpha\beta)^{-1}\boldsymbol{\lambda} + (1 + \alpha\beta)^{-1}\mathbf{h}(\mathbf{z}, \mathbf{x}) \\ &= \begin{bmatrix} -\alpha(1 + \alpha\beta)^{-1}\boldsymbol{\lambda}_1 + (1 + \alpha\beta)^{-1}\mathbf{h}_1 \\ \vdots \\ -\alpha(1 + \alpha\beta)^{-1}\boldsymbol{\lambda}_M + (1 + \alpha\beta)^{-1}\mathbf{h}_M \end{bmatrix} \\ &= [\dot{\boldsymbol{\lambda}}_1^\top \quad \dots \quad \dot{\boldsymbol{\lambda}}_M^\top]^\top. \end{aligned}$$

Therefore, the state and input trajectories computed by iDMPC are equivalent to those computed by iMPC provided that Assumption 3.1 holds. ■

In the next subsection, we exploit the structure of the dynamics of centralised iMPC to directly prove asymptotic convergence of the state $\mathbf{x}$ and predicted trajectories $\mathbf{z}$ of iDMPC.

3.2 Stability analysis

In this subsection, we present the stability analysis of iDMPC. First and foremost, we want to ensure that the state $\mathbf{x}$ of the closed-loop interconnected system (asymptotically) converges to the origin. In addition to the stability of the closed-loop system, we also focus on the behaviour of the predicted trajectories $\mathbf{z}$ computed by iDMPC. Such trajectories must remain uniformly bounded to prevent the controller from malfunctioning. We use the similarity between iDMPC and iMPC that was highlighted in the previous subsection to make conclusions about the stability of iDMPC. The convergence proof is based on analysing QSR dissipativity of the plant. Therefore, we present the following definition.

Definition 3.2 (QSR dissipativity): Let $Q \in \mathbb{R}^{m \times m}$, $S \in \mathbb{R}^{m \times n}$, and $R \in \mathbb{R}^{n \times n}$ be constant matrices. Let

$$w(\mathbf{x}, \mathbf{u}) := \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}^\top \begin{bmatrix} Q & S \\ S^\top & R \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}$$

denote the QSR supply rate. The interconnected system (2) is QSR dissipative if and only if for some storage function $S_x \geq 0$, it holds that $\dot{S}_x \leq w(\mathbf{x}, \mathbf{u})$.

Remark 3.2: Definition 3.2 is equivalent to the classical definition of QSR dissipativity (Antsaklis et al., 2013) for the system output $\mathbf{y} = \mathbf{x}$. When the state $\mathbf{x}$ is observable, it is a straightforward choice for the output $\mathbf{y}$.

We define $E = \text{diag}(E_1, \dots, E_M)$ and note that Assumption 2.2 ensures that there exists a positive constant ν such that $\nabla f(\mathbf{z}, \mathbf{x})^\top \mathbf{z} \geq \nu \mathbf{z}^\top \mathbf{z}$. We are ready to present an important theorem that provides a sufficient condition for asymptotic stability of a dynamical system controlled by iDMPC.

Theorem 3.3: Under Assumption 2.1–3.1, the states $(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ of the closed-loop system and the iDMPC controller depicted

in Figure 1 asymptotically converge to the origin if there exist matrices Q, S, R , and $P = P^\top > 0$ such that

$$\begin{bmatrix} PA_c + A_c^\top P - Q & PB_c - S \\ B_c^\top P - S^\top & -R \end{bmatrix} \leq 0 \quad (20a)$$

$$\begin{bmatrix} -\nu I - \beta H^\top H + E^\top RE & -\beta H^\top V + E^\top S^\top \\ -\beta V^\top H + SE & \frac{1}{4\alpha(1+\alpha\beta)} A^\top A + Q \end{bmatrix} < 0. \quad (20b)$$

Proof: By Theorem 1, the state and input trajectories of iDMPC are equivalent to those of iMPC. Therefore, we can prove asymptotic stability in the centralised setting. We refer to the original proof of stability via dissipativity analysis from Yoshida et al. (2019) and note that Assumption 2.3 ensures that $\boldsymbol{\mu}$ asymptotically converges to zero. ■

Theorem 2 provides a sufficient condition on the asymptotic stability of an interconnected system controlled by iDMPC. However, verifying the sufficient condition given in terms of the LMIs in (20) is not trivial due to the computational complexity. The number of mathematical operations required to check whether a single LMI holds is roughly $O((m+n)^3)$, and thus such computation naturally becomes prohibitive for large m and n . Therefore, we would like to find an alternative method that allows us to verify asymptotic convergence of the states $\mathbf{x}$ and predicted trajectories $\mathbf{z}$ in a distributed fashion. In the following corollary, we present a sufficient condition for asymptotic stability of iDMPC, whose verification has a computational complexity of $O(\sum_{i=1}^M (m_i + n_i)^3)$.

Corollary 3.4: Under Assumption 2.1–3.1, the states $(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ of the closed-loop system and the iDMPC controller asymptotically converge to the origin if there exist matrices $P_i = P_i^\top > 0$, Q_i , S_i , and R_i such that for every $i = 1, \dots, M$,

$$\begin{bmatrix} P_i A_{ci} + A_{ci}^\top P_i - Q_i & P_i B_{ci} - S_i \\ B_{ci}^\top P_i - S_i^\top & -R_i \end{bmatrix} \leq 0 \quad (21a)$$

$$\begin{bmatrix} -\nu I_i - \beta H_i^\top H_i + E_i^\top R_i E_i & -\beta H_i^\top V_i + E_i^\top S_i^\top \\ -\beta V_i^\top H_i + S_i E_i & \frac{1}{4\alpha(1+\alpha\beta)} A_i^\top A_i + Q_i \end{bmatrix} < 0. \quad (21b)$$

Proof: Since the network dynamics are decoupled, the matrices P, Q, S, R, H, V, A, B , and E are block-diagonal. Therefore, (21a) and (21b) together present a sufficient condition for the asymptotic convergence of the local states $(\mathbf{x}_i, \mathbf{z}_i, \boldsymbol{\lambda}_i)$. Since the LMIs are satisfied for all subsystems, then $\boldsymbol{\mu}_i$ asymptotically converges to zero for $i = 1, \dots, M$ by Assumption 2.3. ■

By Corollary 1, we can individually tune the iDMPC algorithm for each subsystem, test whether its states converge to the equilibrium, and finally make a conclusion about the convergence of all states of the interconnected system. The design steps include the characterisation of the QSR dissipativity of the dynamical system, search for a quadratic storage function, and identification of tuning parameters α and β .

Remark 3.3: It is possible to tune local parameters α_i and β_i instead of global parameters α and β . However, we omit this generalisation in our analysis.

The distributed optimal control strategy established in this section is sufficient for the control of systems with constraints that need to be satisfied merely in a steady state. However, iDMPC does not guarantee that the state $\mathbf{x}$ and predicted trajectories $\mathbf{z}$ remain in the feasible set in a transient state. Since iDMPC is computationally efficient, it is in our best interest to prevent any constraint violation in the transient state using a complementary control method that is easy to evaluate. In the next subsection, we introduce an ERG that corrects the converging state and input trajectories generated by iDMPC such that constraints are satisfied in real-time.

3.3 Explicit reference governor

In the previous subsection, we discussed that the control input $\mathbf{u}_i$ applied by iDMPC is optimal and feasible in a steady state, i.e. when $\mathbf{x}_i = \mathbf{0}$. However, the state $\mathbf{x}_i$ can be far away from the origin when a sudden change in the reference is triggered. Whenever such an event occurs, iDMPC can temporarily drive the dynamical system to infeasible states since it does not instantly solve the full optimisation problem in (15). Naturally, we want to ensure feasibility of our optimisation-based controller in a transient state as well.

In this subsection, we introduce a complementary ERG-based control method that ensures the feasibility of the state $\mathbf{x}_i$ in transient whenever iDMPC alone cannot. The ERG is an auxiliary dynamical system that slowly modifies the reference point at which the dynamical system is supposed to operate. The ERG does not solve any optimisation problem, and thus its computational footprint is relatively small compared to iDMPC. This makes the ERG a great candidate for the design of an inexpensive controller that features iDMPC (or similar controller with optimisation-based dynamics) and satisfies constraints in a transient state. The dynamical system controlled by iDMPC and the ERG-based controller is depicted in Figure 3.

The ERG-based control works on the following principle. There is a stabilising controller that ensures feasibility by stabilising the state $\mathbf{x}_i$ around the reference signal modified by the ERG. At the same time, driving the state $\mathbf{x}_i$ to the modified reference allows the ERG to push the modified reference towards the desired reference point at the origin. This facilitates feasibility in transient and, eventually, regulation of the state $\mathbf{x}_i$ and the control input $\mathbf{u}_i$ around the origin. In the remainder of this subsection, we define the ERG-based control system and integrate

it with iDMPC. We analyse the convergence of the modified reference under the joint iDMPC and ERG-based controller.

We let $\mathbf{x}_i^*$ and $\mathbf{u}_i^*$ denote a modified state and input reference of subsystem i . We note that $\mathbf{x}_i^*$ and $\mathbf{u}_i^*$ are not located at the origin in general but it is imperative to drive them to the origin to solve the original regulation problem in (13). In addition to the potentially unsafe iDMPC control input $\mathbf{u}_i$, we introduce a stabilising ERG-based control law

$$\bar{\mathbf{u}}_i = \mathbf{u}_i^* + K_i(\mathbf{x}_i - \mathbf{x}_i^*), \quad (22)$$

where K_i is a constant gain matrix, and the input reference $\mathbf{u}_i^*$ satisfies $A_{ci}\mathbf{x}_i^* + B_{ci}\mathbf{u}_i^* = \mathbf{0}$. We define a Lyapunov function

$$\mathcal{W}_i(\mathbf{x}_i, \mathbf{x}_i^*) = (\mathbf{x}_i - \mathbf{x}_i^*)^T \bar{P}_i (\mathbf{x}_i - \mathbf{x}_i^*), \quad (23)$$

where $\bar{P}_i > 0$. By the standard synthesis of stabilising controllers for linear systems, the gain matrix K_i is chosen such that for some $\bar{P}_i$, it holds that $Q_i = (A_{ci} + B_{ci}K_i)^T \bar{P}_i + \bar{P}_i(A_{ci} + B_{ci}K_i) < 0$. In the design of the ERG dynamics, we take advantage of the set invariance property induced by the stabilising control input $\bar{\mathbf{u}}_i$.

To ensure that no state constraint violation occurs in transient state, we must define a relationship between the set invariance property and the state constraints. We let $\mathcal{N}_i^k$ denote a set of subsystems that are coupled through the k th constraint of subsystem i . We recall from (10) that the constraint function $\mathbf{g}(\mathbf{z}, \mathbf{x})$ is formulated over an N -step horizon, and hence it plays role in the selection of the current control input $\mathbf{u}_i$ as well as the predicted control inputs by iDMPC. In contrast, the ERG facilitates an application of the safe control input $\bar{\mathbf{u}}_i$, which has to satisfy the constraint $\mathbf{q}_{x,i}(\mathbf{x}_i, \mathbf{x}_{-i}) \leq 0$ at the current state. We assume that $\mathbf{q}_{x,i}(\mathbf{x}_i, \mathbf{x}_{-i})$, which is convex by Assumption 2.3, is given by a polyhedron

$$\mathbf{q}_{x,i}(\mathbf{x}_i, \mathbf{x}_{-i}) = L_i \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_{-i} \end{bmatrix} + \boldsymbol{\gamma}_i, \quad (24)$$

where the matrix L_i and the vector $\boldsymbol{\gamma}_i$ contain constant coefficients. Associated with each element of $\mathbf{q}_{x,i}(\mathbf{x}_i, \mathbf{x}_{-i})$ is a vector of Lyapunov functions of subsystem i whose k th element is given by

$$[\mathbf{W}_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*)]_k = \sum_{j \in \mathcal{N}_i^k} \mathcal{W}_j(\mathbf{x}_j, \mathbf{x}_j^*). \quad (25)$$

Assuming that the modified state reference $\mathbf{x}_i^*$ is feasible for all subsystems, there exists a margin between safety and constraint

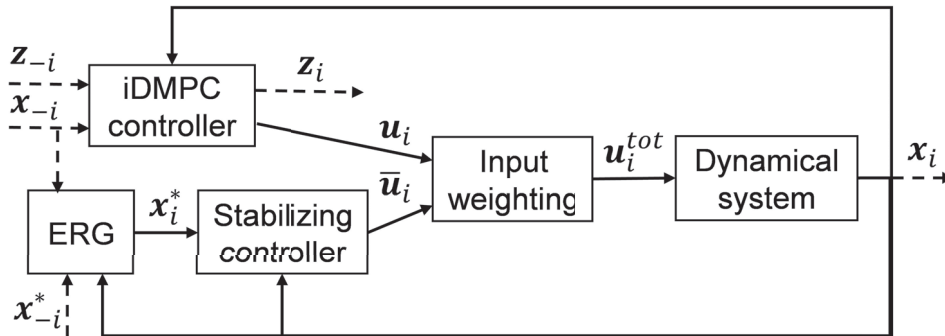


Figure 3. Block diagram of the augmented closed-loop system: iDMPC in the upper loop computes a (sub)optimal input and the stabilising controller with ERG in the lower loop provide a safe stabilising input. The communicated signals are depicted by dashed lines.

violation for $\mathbf{x}_i$. We express the safety margin in terms of Lyapunov level sets, where we use the matrices $\bar{P}_i$ that certified asymptotic stability of $\mathbf{x}_i$ under the control input $\bar{\mathbf{u}}_i$. The level set associated with the k th constraint of subsystem i defined in (24) is given as follows Nicotra and Garone (2015)

$$[\Gamma_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k = \frac{\|[\mathbf{q}_{x,i}(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k\|^2}{[L_i]_k \bar{P}_{\mathcal{N}_i}^{-1} [L_i]_k^\top}, \quad (26)$$

where $\bar{P}_{\mathcal{N}_i} = \text{blkdiag}(\bar{P}_i, \bar{P}_{-i})$. We succinctly write $\Gamma(\mathbf{x}^*) = [\Gamma_1^\top \dots \Gamma_M^\top]^\top$ and $\mathbf{W}(\mathbf{x}, \mathbf{x}^*) = [\mathbf{W}_1^\top \dots \mathbf{W}_M^\top]^\top$. Finally, we define a set $\mathcal{S}$ such that

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^m, \mathbf{W}(\mathbf{x}, \mathbf{x}^*) \leq \Gamma(\mathbf{x}^*)\}. \quad (27)$$

The set $\mathcal{S}$ is depicted in Figure 4, which illustrates the basic principle of the ERG. The modified reference $\mathbf{x}^*$ is meant to be driven to the origin while the state $\mathbf{x}$ remains sufficiently close so that it does not violate the safety margin $\Gamma(\mathbf{x}^*)$. In the following proposition, we show that the set $\mathcal{S}$ is safe under the stabilising control $\bar{\mathbf{u}} = [\bar{\mathbf{u}}_1^\top \dots \bar{\mathbf{u}}_M^\top]^\top$.

Proposition 3.5: *For every feasible reference $\mathbf{x}^*$, the state $\mathbf{x} \in \mathcal{S}$ is feasible and the set $\mathcal{S}$ is invariant under the stabilising control input $\bar{\mathbf{u}}$.*

Proof: Since the modified reference $\mathbf{x}^*$ is feasible, then $\Gamma(\mathbf{x}^*) \geq 0$. Furthermore, under the asymptotically stabilising control input $\bar{\mathbf{u}}$, we have for each subsystem

$$\dot{\mathcal{V}}_i(\mathbf{x}_i, \mathbf{x}_i^*) = (\mathbf{x}_i - \mathbf{x}_i^*)^\top Q_i (\mathbf{x}_i - \mathbf{x}_i^*) \leq -c_i \|\mathbf{x}_i - \mathbf{x}_i^*\|^2,$$

where c_i is the largest eigenvalue of Q_i . Therefore, $\dot{\mathbf{W}} \leq 0$ and the set $\mathcal{S}$ is invariant. ■

Proposition 3.5 stipulates that the constraints cannot be violated when the state $\mathbf{x}$ is sufficiently close to a fixed modified reference $\mathbf{x}^*$ and the system is controlled by the stabilising input $\bar{\mathbf{u}}$. However, the modified reference $\mathbf{x}^*$ must not remain fixed unless it is located at the origin.

In the following lines, we design the ERG dynamics to drive the modified reference $\mathbf{x}^*$ to the origin and to ensure safety of the state $\mathbf{x}$ in the process. We define a gain margin for subsystem i as follows

$$\Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*) = \min_k [\Gamma_i - \mathbf{W}_i]_k. \quad (28)$$

Furthermore, we let $\rho_i(\mathbf{x}_i^*) = \text{sign}(\mathbf{x}_i^*)$ denote a vector field, where $\text{sign}(\cdot)$ is a sign function. Additionally, we introduce a

vector function $\xi_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)$ whose k th element is given by

$$[\xi_i]_k = \begin{cases} \min \left\{ 1, \frac{[\Gamma_i]_k - \epsilon}{\epsilon} \right\} & \text{if } \nabla_{\mathbf{x}_i^*} [\Gamma_i]_k \cdot [\rho_i(\mathbf{x}_i^*)]_k < 0 \\ 1 & \text{if } \nabla_{\mathbf{x}_i^*} [\Gamma_i]_k \cdot [\rho_i(\mathbf{x}_i^*)]_k \geq 0 \end{cases}$$

for some small $\epsilon > 0$. Finally, the ERG dynamics are given as

$$\dot{\mathbf{x}}_i^* = -\sigma \cdot \hat{\xi}_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) \cdot \Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*) \cdot \rho_i(\mathbf{x}_i^*), \quad (29)$$

where $\sigma > 0$ and $\hat{\xi}_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) = \min_k [\xi_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k$. Each term in the ERG dynamics has a specific purpose. We clarify that in the following lines.

- (1) The constant σ controls the convergence rate.
- (2) The term $\hat{\xi}_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)$ ensures that the modified reference $\mathbf{x}_i^*$ does not get too close to the boundary of the feasible set. This prevents slow convergence of $\mathbf{x}_i^*$ and convergence to a point on the boundary of the feasible set.
- (3) The gain margin $\Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*)$ controls the convergence rate; it decreases with an increasing distance between $\mathbf{x}$ and $\mathbf{x}^*$.
- (4) The vector field $\rho_i(\mathbf{x}_i^*)$ provides a direction to the origin.

Furthermore, the gain margin $\Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*)$ is useful in the control synthesis because it provides a metric, based on which the subsystems can transition from the performance-based iDMPC control input $\mathbf{u}_i$ to the safe control input $\bar{\mathbf{u}}_i$. We propose a smooth control law as follows

$$\mathbf{u}_i^{\text{tot}} = \theta_i \mathbf{u}_i + (1 - \theta_i) \bar{\mathbf{u}}_i, \quad (30)$$

where $\theta_i = \min\{\varphi_i \Delta_i, 1\}$ and $\varphi_i > 0$ is a parameter that controls the smoothness of the transition between the safe and suboptimal control inputs. We make the following assumption on the control input $\mathbf{u}_i^{\text{tot}}$.

Assumption 3.2: *The control input $\mathbf{u}_i^{\text{tot}}$ is always feasible, i.e. $\mathbf{u}_i^{\text{tot}} \in \mathcal{U}_i$.*

Assumption 3.2 is important for us to be able to stabilise each subsystem around the modified reference $\mathbf{x}_i^*$, and therefore ensure its feasibility. We define a set $\mathcal{S}' = \{\mathbf{x} \in \mathbb{R}^m, \mathbf{W}(\mathbf{x}, \mathbf{x}^*) \leq \Gamma'(\mathbf{x}^*)\}$ for some $\Gamma'(\mathbf{x}^*) < \Gamma(\mathbf{x}^*)$, and introduce an important lemma that certifies convergence and safety under the control input $\mathbf{u}_i^{\text{tot}}$.

Lemma 3.6: *For every fixed strictly feasible reference $\mathbf{x}^*$, the state $\mathbf{x} \in \mathcal{S}$ is feasible and the set $\mathcal{S}$ is invariant under control input $\mathbf{u}_i^{\text{tot}}$, $i = 1, \dots, M$. Furthermore, $\mathbf{x}$ asymptotically converges to the limit set $\mathcal{S}'$.*

Proof: The proof is analogous to the proof of Proposition 1. subsystem i dissipates energy as follows

$$\dot{\mathcal{V}}_i(\mathbf{x}_i, \mathbf{x}_i^*) \leq -c_i \|\mathbf{x}_i - \mathbf{x}_i^*\|^2 + 2\theta_i (\mathbf{x}_i - \mathbf{x}_i^*)^\top \bar{P}_i B_{ci} (\mathbf{u}_i - \bar{\mathbf{u}}_i).$$

We recall that the control input computed by the iDMPC controller is asymptotically stable, and hence $\|\mathbf{u}_i\|^2 < \infty$. Furthermore, the stabilising controller satisfies $\|\bar{\mathbf{u}}_i\|^2 < \infty$ for any

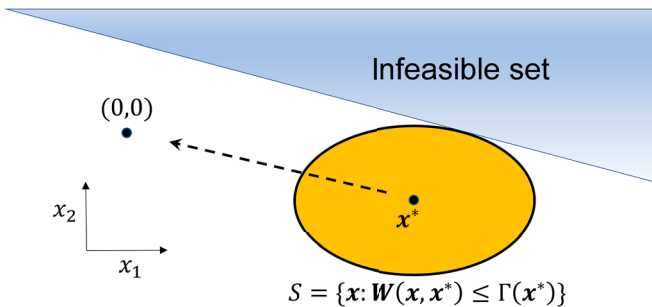


Figure 4. Spatial representation of the invariant set $\mathcal{S}$ (orange) whose interior contains feasible states $\mathbf{x}$.

strictly feasible reference $\mathbf{x}_i^*$ and bounded state $\mathbf{x}_i$. Hence, for $c'_i = \sup_{\mathbf{u}_i, \bar{\mathbf{u}}_i} \|\bar{P}_i B_{ci}(\mathbf{u}_i - \bar{\mathbf{u}}_i)\|^2 < \infty$, we obtain

$$\dot{\mathcal{V}}_i(\mathbf{x}_i, \mathbf{x}_i^*) \leq -c_i \|\mathbf{x}_i - \mathbf{x}_i^*\|^2 + 2\theta_i c'_i \|\mathbf{x}_i - \mathbf{x}_i^*\|^2.$$

We recognise that the state $\mathbf{x}_i$ asymptotically converges for $\theta_i = \min\{1, \varphi_i \Delta_i\} = \min\{1, \varphi_i \min_k [\Gamma_i - \mathbf{W}_i]_k\} < \frac{c_i}{2c'_i}$. This condition is satisfied if $\min_k [\Gamma_i - \mathbf{W}_i]_k \in [0, \frac{c_i}{2\varphi_i c'_i}]$. We note that $\Gamma_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) > 0$ since the modified reference $\mathbf{x}^*$ is strictly feasible. Letting $[\Gamma'_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k = \max\{0, [\Gamma_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k - \frac{c_i}{2\varphi_i c'_i}\}$, it holds that $\Gamma'_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) < \Gamma_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)$ and we obtain $\dot{\mathcal{V}}_i(\mathbf{x}_i, \mathbf{x}_i^*) \leq 0$ when $\min_k [\Gamma'_i - \mathbf{W}_i]_k = 0$. Therefore, the state of the interconnected system $\mathbf{x}$ asymptotically converges to the limit set $\mathcal{S}'$. ■

Lemma 3.6 presents an intermediate result in our analysis that suggests that the state $\mathbf{x}$ asymptotically converges to the interior of the largest safe Lyapunov set when the subsystems apply the control law $\mathbf{u}_i^{tot}$ and the modified reference $\mathbf{x}_i^*$ remains fixed. It is important to note that the potentially unsafe iDMPC control input $\mathbf{u}_i$ does not compromise the safety of our controller. In fact, the safety is not compromised even for a moving modified reference $\mathbf{x}_i^*$ as the gain margin $\Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*)$ controls its convergence rate. Since we design a controller that does not have the computational power to run full optimisation and enough memory to store precomputed control sequences, it is reasonable to assume that feasibility can be achieved by stabilising the subsystem around the feasible modified reference $\mathbf{x}_i^*$ (Garone & Nicotra, 2015). As a final step in our analysis, we need to show that the modified reference $\mathbf{x}_i^*$ asymptotically converges to the origin for $i = 1, \dots, M$.

Theorem 3.7: Suppose that the initial reference $\mathbf{x}^*$ is strictly feasible and the initial state satisfies $\mathbf{x} \in \mathcal{S}$. Under Assumption 2.1–3.2, the modified reference $\mathbf{x}^*$ asymptotically converges to the origin under the ERG dynamics in (29) and the control law in (30).

Proof: We let $\mathcal{V}(\mathbf{x}^*) = \sum_{i=1}^M \frac{1}{2} \|\mathbf{x}_i^*\|^2$ denote the Lyapunov function for the ERG. Its time derivative is given as

$$\begin{aligned} \dot{\mathcal{V}}(\mathbf{x}^*) &= -\sigma \sum_{i=1}^M \hat{\xi}_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) \cdot \Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*) \cdot \mathbf{x}_i^{*\top} \boldsymbol{\rho}_i(\mathbf{x}_i^*) \\ &= -\sigma \sum_{i=1}^M \hat{\xi}_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) \cdot \Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*) \cdot \|\mathbf{x}_i^*\|_1. \end{aligned}$$

Since the initial state $\mathbf{x}$ and modified reference $\mathbf{x}^*$ are feasible, the ERG dynamics ensure that $\Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*) \geq 0$ and $\hat{\xi}_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) \geq 0$ for $i = 1, \dots, M$, assuming that ϵ is sufficiently small. We apply the LaSalle's invariance principle to prove asymptotic stability of the origin. If the term $\Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*) = 0$ at $t = T_1$, then the state $\mathbf{x}_i$ converges to the modified reference $\mathbf{x}_i^*$ since $[\Gamma_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k \geq \epsilon$. Hence, there exists $T_2 > T_1$ such that $\Delta_i(\mathbf{x}_i, \mathbf{x}_i^*, \mathbf{x}_{-i}, \mathbf{x}_{-i}^*) > 0$ at $t = T_2$. Moreover, the term $\hat{\xi}_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) = 0$ at $t = T_1$ only when $[\Gamma_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k = \epsilon$ and $\nabla_{\mathbf{x}_i^*} [\Gamma_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k \cdot [\boldsymbol{\rho}_i(\mathbf{x}_i^*)]_k < 0$ for

some k . Since the constraints are convex by Assumption 2.3, then there exists $T_2 > T_1$ such that $[\Gamma_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*)]_k > \epsilon$, and consequently $\hat{\xi}_i(\mathbf{x}_i^*, \mathbf{x}_{-i}^*) > 0$, for $t = T_2$. Therefore, the modified reference $\mathbf{x}^*$ converges to the desired reference at the origin. ■

Having proved that the control input $\mathbf{u}_i^{tot}$ ensures safety and the modified reference $\mathbf{x}_i^*$ converges to the origin, we essentially established that our proposed control design solves the original regulation problem. However, it is obvious that the proposed control $\mathbf{u}_i^{tot}$ is very conservative because it prioritises safety before performance. The trade-off between performance and safety is inevitable while operating on a constrained budget. The empirical results in the next section are consistent with our claim. In the simulations, we compare iDMPC and ERG when applied alone, as well as the joint iDMPC-ERG control and benchmark DMPC. We focus on the performance, feasibility, and computational footprint of these control methods.

4. Simulation results

In this section, we present simulation results of our proposed safe iDMPC controller. We consider three subsystems modeled as double integrators. The subsystems are driven to the state reference $\mathbf{r}_{x_1} = [0 \ 0]^\top$, $\mathbf{r}_{x_2} = [5 \ 0]^\top$ and $\mathbf{r}_{x_3} = [10 \ 0]^\top$. The subsystems must satisfy the following constraints: $\|[\mathbf{x}_i]_2\| \leq 10$, $[\mathbf{x}_1]_1 - [\mathbf{x}_2]_1 \leq 0$, and $[\mathbf{x}_2]_1 - [\mathbf{x}_3]_1 \leq 0$. Furthermore, each subsystem minimises a cost function $f_i(\mathbf{z}_i, \mathbf{z}_{-i}, \mathbf{x}_i, \mathbf{x}_{-i}) = \sum_{k=0}^N c_i(\mathbf{x}_i^{k\Delta t}, \mathbf{x}_{-i}^{k\Delta t}, \mathbf{u}_i^{k\Delta t})$, where $c_i(\mathbf{x}_i, \mathbf{x}_{-i}, \mathbf{u}_i) = \|\mathbf{u}_i - \mathbf{r}_{u_i}\|^2 + 3\|\mathbf{x}_i - \mathbf{r}_{x_i}\|^2 + \sum_{(i,j) \in \mathcal{E}} (\mathbf{x}_i - \mathbf{r}_{x_i})^\top (\mathbf{x}_j - \mathbf{r}_{x_j})$. The reference $\mathbf{r}_{u_i}$ satisfies $A_{ci}\mathbf{r}_{x_i} + B_{ci}\mathbf{r}_{u_i} = 0$. We compute the predicted trajectories with $\Delta t = 0.2s$, and $N = 5$. In the simulations, we consider:

- (1) DMPC - the results are obtained using the Matlab function *quadprog*;
- (2) iDMPC - each subsystem is pre-stabilised around the state reference by applying a linear state feedback $\mathbf{u}_i^{pre} = \mathbf{r}_{u_i} + K_i(\mathbf{x}_i - \mathbf{r}_{x_i})$, where $K_1 = [-2 \ -3]$, $K_2 = [-4 \ -4]$, and $K_3 = [-5 \ -3]$, respectively. The iDMPC parameters that ensure the asymptotic stability are $\alpha_1 = \alpha_2 = \beta_2 = \beta_3 = 10$, $\beta_1 = 5$, and $\alpha_3 = 1000$;
- (3) ERG - each subsystem is stabilised around the modified reference $\mathbf{x}_i^*$ with the safe control $\bar{\mathbf{u}}_i$, where the feedback gains are K_1, K_2 , and K_3 ; furthermore, the ERG hyperparameters are $\sigma = 1$ and $\epsilon = 10^{-3}$;
- (4) iDMPC/ERG - the results are obtained with the iDMPC and ERG parameters, and the smoothing parameters are $\varphi_1 = \varphi_2 = \varphi_3 = 0.1$.

The algorithms are implemented in *Intel Core i7-6700HQ CPU @2.6 GHz*. We use the ODE45 solver to emulate the analog computation of iDMPC, ERG, and iDMPC/ERG in Simulink. In Figure 5, we compare the positions $[\mathbf{x}_i]_1$ of the systems controlled by DMPC, iDMPC, ERG, and iDMPC/ERG. We assume that DMPC computes the entire optimisation problem instantly at every time step. As we expected, DMPC is superior when we compare the performance and safety of the above-mentioned algorithms. Although iDMPC control shows similar performance (Figure 6), the trajectories violate safety constraints in

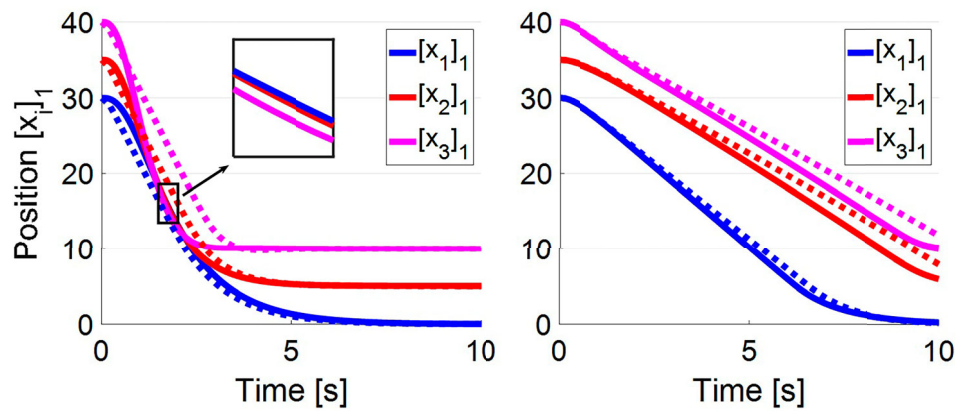


Figure 5. The first plot shows the position of subsystems controlled by iDMPC (solid line) and DMPC (dashed line). The second plot depicts the position of subsystems controlled by iDMPC/ERG (solid) and ERG (dashed).

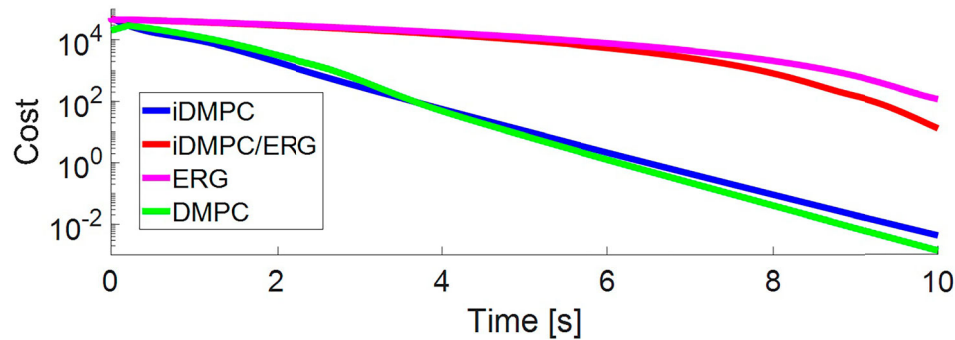


Figure 6. The cost incurred with iDMPC, iDMPC/ERG, ERG, and DMPC controllers.

transient (Figure 5, left). In contrast, the ERG and iDMPC/ERG schemes provide safety guarantees at a higher overall cost. The total computational times are 0.2594 s for DMPC, 0.130 s for iDMPC, and 0.192 s for iDMPC/ERG. The iDMPC controller is also faster than its centralised counterpart iMPC, whose computational time is 0.167 s.

5. Conclusion and future work

Instant distributed MPC (iDMPC) is a computationally efficient optimal control method that quickly updates predicted input and state trajectories. We showed that iDMPC essentially performs the same input updates as its centralised counterpart, iMPC, in the absence of time delays in the communication. Even though iDMPC alone does not guarantee the feasibility of the state and input trajectories in transient, we showed that it can be augmented by an ERG to do exactly that. We combined the strengths of iDMPC and ERG, namely performance and safety, to establish a new computationally efficient control method. We note that the scheme with iDMPC and the ERG-based control generalises to other classes of optimisation dynamics beyond the primal-dual algorithm (Yamashita et al., 2020). There are a few challenges that need to be addressed in future work. One, the desired reference is usually a lower-dimensional signal. Therefore, further research efforts should focus on the extension to output reference tracking and underlying dissipativity-based stability guarantees. Two, it should be investigated if the iDMPC scheme can be tuned to ensure convergence and feasibility with tighter communication constraints. The current

assumption that subsystems exchange data with no delays is rather strong. Dissipativity could play an important role in the stability analysis under time delays.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work of Vijay Gupta and Martin Figura was partially supported by NSF Award 2300355, ARO Award 203708PU and AFOSR award 204152PU. The work of Masaki Inoue was partially supported by Grant-in-Aid for Scientific Research (B), No. ~20H02173 from JSPS.

References

- Adegbege, A. A. (2022). Constrained continuous-time dynamics for linear model predictive control. *IEEE Control Systems Letters*, 6(1), 3098–3103. <https://doi.org/10.1109/LCSYS.2022.3180890>
- Antsaklis, P. J., Goodwine, B., Gupta, V., McCourt, M. J., Wang, Y., Wu, P., Xia, M., Yu, H., & Zhu, F. (2013). Control of cyberphysical systems using passivity and dissipativity based methods. *European Journal of Control*, 19(5), 379–388. <https://doi.org/10.1016/j.jejcon.2013.05.018>
- Bemporad, A. (1998). Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3), 415–419. <https://doi.org/10.1109/9.661611>
- Camponogara, E., Jia, D., Krogh, B. H., & Talukdar, S. (2002). Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1), 44–52. <https://doi.org/10.1109/37.980246>
- Casavola, A., Garone, E., & Tedesco, F. (2014). A distributed multi-agent command governor strategy for the coordination of networked interconnected systems. *IEEE Transactions on Automatic Control*, 59(8), 2099–2112. <https://doi.org/10.1109/TAC.2014.2318081>

- Dunbar, W. B., & Murray, R. M. (2006). Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4), 549–558. <https://doi.org/10.1016/j.automatica.2005.12.008>
- Figura, M., Su, L., Gupta, V., & Inoue, M. (2020). Instant distributed model predictive control for constrained linear systems. In *2020 American Control Conference* (pp. 4582–4587). IEEE.
- Garone, E., Di Cairano, S., & Kolmanovsky, I. (2017). Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75(1), 306–328. <https://doi.org/10.1016/j.automatica.2016.08.013>
- Garone, E., Nicotra, M., & Ntogramatzidis, L. (2018). Explicit reference governor for linear systems. *International Journal of Control*, 91(6), 1415–1430. <https://doi.org/10.1080/00207179.2017.1317832>
- Garone, E., & Nicotra, M. M. (2015). Explicit reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 61(5), 1379–1384. <https://doi.org/10.1109/TAC.2015.2476195>
- Gilbert, E. G., Kolmanovsky, I., & Tan, K. T. (1995). Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *International Journal of Robust and Nonlinear Control*, 5(5), 487–504. [https://doi.org/10.1002/\(ISSN\)1099-1239](https://doi.org/10.1002/(ISSN)1099-1239)
- Hauswirth, A., Bolognani, S., Hug, G., & Dörfler, F. (2020). Timescale separation in autonomous optimization. *IEEE Transactions on Automatic Control*, 66(2), 611–624. <https://doi.org/10.1109/TAC.9>
- Jokic, A., & Lazar, M. (2009). On constrained steady-state regulation: Dynamic KKT controllers. *IEEE Transactions on Automatic Control*, 54(9), 2250–2254. <https://doi.org/10.1109/TAC.2009.2026856>
- Keviczky, T., Borrelli, F., & Balas, G. J. (2006). Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12), 2105–2115. <https://doi.org/10.1016/j.automatica.2006.07.008>
- Kolmanovsky, I., Garone, E., & Di Cairano, S. (2014). Reference and command governors: A tutorial on their theory and automotive applications. In *2014 American Control Conference* (pp. 226–241). IEEE.
- Leirens, S., Zamora, C., Negenborn, R., & De Schutter, B. (2010). Coordination in urban water supply networks using distributed model predictive control. In *2010 American Control Conference* (pp. 3957–3962). IEEE.
- Liao-McPherson, D., Nicotra, M. M., & Kolmanovsky, I. (2020). Time-distributed optimization for real-time model predictive control: Stability, robustness, and constraint satisfaction. *Automatica*, 117(1), Article 108973. <https://doi.org/10.1016/j.automatica.2020.108973>
- Limón, D., Alvarado, I., Alamo, T., & Camacho, E. F. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9), 2382–2387. <https://doi.org/10.1016/j.automatica.2008.01.023>
- Liu, A., Zhang, W. A., Yu, L., Yan, H., & Zhang, R. (2018). Formation control of multiple mobile robots incorporating an extended state observer and distributed model predictive approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(11), 4587–4597. <https://doi.org/10.1109/TSMC.6221021>
- Negenborn, R. R., & Maestre, J. M. (2014). Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, 34(4), 87–97. <https://doi.org/10.1109/MCS.2014.2320397>
- Nicotra, M. M., Bartulovic, M., Garone, E., & Sinopoli, B. (2015). A distributed explicit reference governor for constrained control of multiple UAVs. *IFAC-PapersOnLine*, 48(22), 156–161. <https://doi.org/10.1016/j.ifacol.2015.10.323>
- Nicotra, M. M., & Garone, E. (2015). Explicit reference governor for continuous time nonlinear systems subject to convex constraints. In *2015 American Control Conference* (pp. 4561–4566). IEEE.
- Nicotra, M. M., Liao-McPherson, D., & Kolmanovsky, I. V. (2018). Embedding constrained model predictive control in a continuous-time dynamic feedback. *IEEE Transactions on Automatic Control*, 64(5), 1932–1946. <https://doi.org/10.1109/TAC.9>
- Patel, N. R., Risbeck, M. J., Rawlings, J. B., Wenzel, M. J., & Turney, R. D. (2016). Distributed economic model predictive control for large-scale building temperature regulation. In *2016 American Control Conference* (pp. 895–900). IEEE.
- Pourkargar, D. B., Almansoori, A., & Daoutidis, P. (2017). Impact of decomposition on distributed model predictive control: A process network case study. *Industrial & Engineering Chemistry Research*, 56(34), 9606–9616. <https://doi.org/10.1021/acs.iecr.7b00644>
- Richards, A., & How, J. P. (2007). Robust distributed model predictive control. *International Journal of Control*, 80(9), 1517–1531. <https://doi.org/10.1080/00207170701491070>
- Richter, S., Jones, C. N., & Morari, M. (2009). Real-time input-constrained MPC using fast gradient methods. In *48th IEEE conference on decision and control held jointly with 28th chinese control conference* (pp. 7387–7393). IEEE.
- Rubagotti, M., Barcelli, D., & Bemporad, A. (2014). Robust explicit model predictive control via regular piecewise-affine approximation. *International Journal of Control*, 87(12), 2583–2593. <https://doi.org/10.1080/00207179.2014.935958>
- Skibik, T., & Nicotra, M. M. (2022). Analysis of time-distributed model predictive control when using a regularized primal–Dual gradient optimizer. *IEEE Control Systems Letters*, 7(1), 235–240. <https://doi.org/10.1109/LCSYS.2022.3186631>
- Stewart, B. T., Venkat, A. N., Rawlings, J. B., Wright, S. J., & Pannocchia, G. (2010). Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8), 460–469. <https://doi.org/10.1016/j.sysconle.2010.06.005>
- Tedesco, F., Casavola, A., & Garone, E. (2012). Distributed command governor strategies for constrained coordination of multi-agent networked systems. In *2012 American Control Conference* (pp. 6005–6010). IEEE.
- Tøndel, P., Johansen, T. A., & Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3), 489–497. [https://doi.org/10.1016/S0005-1098\(02\)00250-9](https://doi.org/10.1016/S0005-1098(02)00250-9)
- Venkat, A. N., Hiskens, I. A., Rawlings, J. B., & Wright, S. J. (2008). Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6), 1192–1206. <https://doi.org/10.1109/TCST.2008.919414>
- Wang, Y., & Boyd, S. (2009). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278. <https://doi.org/10.1109/TCST.2009.2017934>
- Yamashita, S., Hatanaka, T., Yamauchi, J., & Fujita, M. (2020). Passivity-based generalization of primal–dual dynamics for non-strictly convex cost functions. *Automatica*, 112(1), Article 108712. <https://doi.org/10.1016/j.automatica.2019.108712>
- Yildirim, E. A., & Wright, S. J. (2002). Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization*, 12(3), 782–810. <https://doi.org/10.1137/S1052623400369235>
- Yoshida, K., Inoue, M., & Hatanaka, T. (2019). Instant MPC for linear systems and dissipativity-based stability analysis. *IEEE Control Systems Letters*, 3(4), 811–816. <https://doi.org/10.1109/LCSYS.7782633>